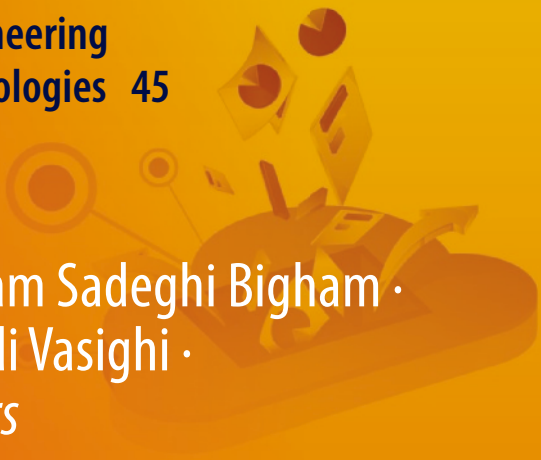
Lecture Notes on Data Engineering
and Communications Technologies 45



Mahdi Bohlouli · Bahram Sadeghi Bigham ·
Zahra Narimani · Mahdi Vasighi ·
Ebrahim Ansari *Editors*

Data Science: From Research to Application

Springer

Lecture Notes on Data Engineering and Communications Technologies

Volume 45

Series Editor

Fatos Xhafa, Technical University of Catalonia, Barcelona, Spain

mmeybodi@aut.ac.ir

The aim of the book series is to present cutting edge engineering approaches to data technologies and communications. It will publish latest advances on the engineering task of building and deploying distributed, scalable and reliable data infrastructures and communication systems.

The series will have a prominent applied focus on data technologies and communications with aim to promote the bridging from fundamental research on data science and networking to data engineering and communications that lead to industry products, business knowledge and standardisation.

**** Indexing: The books of this series are submitted to SCOPUS, ISI Proceedings, MetaPress, Springerlink and DBLP ****

More information about this series at http://www.springer.com/series/15362

mmeybodi@aut.ac.ir

Mahdi Bohlouli · Bahram Sadeghi Bigham ·
Zahra Narimani · Mahdi Vasighi ·
Ebrahim Ansari
Editors

Data Science: From Research to Application

Springer

mmeybodi@aut.ac.ir

Editors
Mahdi Bohlouli
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

Bahram Sadeghi Bigham
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

Zahra Narimani
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

Mahdi Vasighi
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

Ebrahim Ansari
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

ISSN 2367-4512 ISSN 2367-4520 (electronic)
Lecture Notes on Data Engineering and Communications Technologies
ISBN 978-3-030-37308-5 ISBN 978-3-030-37309-2 (eBook)
https://doi.org/10.1007/978-3-030-37309-2

© Springer Nature Switzerland AG 2020
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

mmeybodi@aut.ac.ir

Preface

Data science is a rapidly growing field and as a profession incorporates a wide variety of areas from statistics, mathematics and machine learning to applied big data analytics. This is not limited to computer science, but also physics, astronomy, medicine, labour market analysis, marketing and many more. For instance, the Large Synoptic Survey Telescope (LSST) is planned to deliver 100 petabytes of data in the next decade12. This demands awareness of emerging technologies and calls for technologies and services of tomorrow, especially big data and data science expertise. According to the Forbes, "Data Science" is listed as LinkedIn's fastest growing job in 2017. The need to have professional data scientists is not limited to just education, but also proper development and preparation of environments and tools to brainstorm on recent research and scientific achievements of the area.

Such a scientific need for having an event that scientists can share their first-class data science achievements was known for us since years ago, and we tried in this regard to put special interest and focus on data science in our past executions of CICIS conference. This year, in its fifth run, we decided to dedicate the conference to data science area and accordingly to even keep it as professional data science event in the future. The 5th international conference on Contemporary issues in Data Science (CiDaS) has provided a sort of a real workshop (not listen-shop) to scientists and scholars to share ideas, initiative future collaborations and brainstorm challenges as well as industries to catch emerging solutions from the science to their real data science problems. In this regard, we tried hard in the frame of CiDaS 2019 to support all scientists and involve them in this move towards the successful future. In addition, we received special general interest from most of academics and industries in Iran and abroad by submitting significant number of manuscripts.

In particular, we received submissions from ten different countries and have tried to deliver at least two constructive reviews per submission. The acceptance rate of full paper submissions in CiDaS 2019 was about 30%. Furthermore, CiDaS 2019 has had significant number of national and international sponsors. Current chapters of this book are accepted papers of the conference. In addition, we also had scholarly well-known keynote speakers, who covered wide range of data science topics from academic and industrial points of views. By having over 55 experts as

mmeybodi@aut.ac.ir

scientific committee members from over 15 countries, we provided multinational and context-aware reviews to our audience, which also improved the quality of accepted papers as well.

We believe that we will be able to support all data scientists from various areas in our future events and involve them in this move towards the successful future and welcome your support. We hope that you will enjoy our future iterations of CiDaS. If you find our work interesting for you and your field, we always welcome collaborations and supports in this scientific event.

Sincerely Yours,

Mahdi Bohlouli
Bahram Sadeghi Bigham
Zahra Narimani
Mahdi Vasighi
Ebrahim Ansari
CiDaS 2019 Steering Committee

mmeybodi@aut.ac.ir

Acknowledgement

We would like to appreciate all scientific supports of following scholars through their reviews and constructive feedback:

- Hassan Abolhassani, Software Engineer, Google, USA
- Mohsen Afsharchi, University of Zanjan, Iran
- Sadegh Aliakbary, Shahid Beheshti University, Iran
- Ali Amiri, University of Zanjan, Iran
- Morteza AnaLoui, Iran University of Science and Technology, Iran
- Amin Anjomshoa, Massachusetts Institute of Technology, USA
- Lefteris Angelis, Aristotle University of Thessaloniki, Greece
- Nikos Askitas, Research Data Center, Institute of Labour Economics, Germany
- Zeinab Bahmani, Uni-Select Inc., Canada
- Davide Ballabio, University of Milano-Bicocca, Italy
- Markus Bick, ESCP Europe Business School, Germany
- Elnaz Bigdeli, University of Ottawa, Canada
- Mansoor Davoodi Monfared, Institute of Advanced Studies in Basic Sciences, Iran
- Mohammad Reza Faraji, Institute of Advanced Studies in Basic Sciences, Iran
- Agata Filipowska, Poznan University of Economics and Business, Poland
- Holger Fröhlich, University of Bonn, Germany
- George Kakarontzas, Technical Educational Institute of Thessaly, Greece
- Alireza Khastan, Institute of Advanced Studies in Basic Sciences, Iran
- Antonio Liotta, University of Derby, UK
- Rahim Mahmoudvand, Bu-Ali Sina University, Iran
- Samaneh Mazaheri, University of Ontario Institute of Technology, Canada
- Federico Marini, University of Rome "La Sapienza", Italy
- Maryam Mehri Dehnavi, University of Toronto, Canada
- Nima Mirbakhsh, Arcane Inc., Canada
- Ali Movaghar, Sharif University of Technology, Iran
- Ehsan Nedaaee Oskoee, Institute of Advanced Studies in Basic Sciences, Iran
- Peyman Pahlevani, Institute of Advanced Studies in Basic Sciences, Iran

mmeybodi@aut.ac.ir

- Paurush Praveen, Machine Learning Research, CluePoints, Belgium
- Edy Portmann, University of Fribourg, Switzerland
- Masoud Rahgozar, University of Tehran, Iran
- Shahram Rahimi, Southern Illinois University, USA
- Reinhard Rapp, Hochschule Magdeburg, Germany
- Mohammad Saraee, University of Salford, Manchester, UK
- Frank Schulz, SAP AG, Germany
- Mehdi Sheikhalishahi, InnoTec21 GmbH, Germany
- Ioannis Stamelos, Aristotle University of Thessaloniki, Greece
- Athena Vakali, Aristotle University of Thessaloniki, Greece

mmeybodi@aut.ac.ir

Contents

Efficient Cluster Head Selection Using the Non-linear Programming Method for Wireless Sensor Networks 1
Maryam Afshoon, Amin Keshavazi, Tajedin Darikvand, and Mahdi Bohlouli

A Survey on Measurement Metrics for Shape Matching Based on Similarity, Scaling and Spatial Distance 13
Bahram Sadeghi Bigham and Samaneh Mazaheri

Static Signature-Based Malware Detection Using Opcode and Binary Information 24
Azadeh Jalilian, Zahra Narimani, and Ebrahim Ansari

RSS_RAID a Novel Replicated Storage Schema for RAID System 36
Saeid Pashazadeh, Leila Namvari Tazehkand, and Reza Soltani

A New Distributed Ensemble Method with Applications to Machine Learning 44
Saeed Taghizadeh, Mahmood Shabankhah, Ali Moeini, and Ali Kamandi

A Glance on Performance of Fitness Functions Toward Evolutionary Algorithms in Mutation Testing 59
Reza Ebrahimi Atani, Hasan Farzaneh, and Sina Bakhshayeshi

Density Clustering Based Data Association Approach for Tracking Multiple Targets in Cluttered Environment 76
Mousa Nazari and Saeid Pashazadeh

Representation Learning Techniques: An Overview 89
Hassan Khastavaneh and Hossein Ebrahimpour-Komleh

A Community Detection Method Based on the Subspace Similarity of Nodes in Complex Networks 105
Mehrnoush Mohammadi, Parham Moradi, and Mahdi Jalili

mmeybodi@aut.ac.ir

Forecasting Multivariate Time-Series Data Using LSTM and Mini-Batches 121
Athar Khodabakhsh, Ismail Ari, Mustafa Bakır, and Serhat Murat Alagoz

Identifying Cancer-Related Signaling Pathways Using Formal Methods 130
Fatemeh Mansoori, Maseud Rahgozar, and Kaveh Kavousi

Predicting Liver Transplantation Outcomes Through Data Analytics 142
Bahareh Kargar, Vahid Gheshlaghi Gazerani, and Mir Saman Pishvaee

Deep Learning Prediction of Heat Propagation on 2-D Domain via Numerical Solution 161
Behzad Zakeri, Amin Karimi Monsefi, and Babak Darafarin

Cluster Based User Identification and Authentication for the Internet of Things Platform 175
Rafflesia Khan and Md.Rafiqul Islam

Forecasting of Customer Behavior Using Time Series Analysis 188
Hossein Abbasimehr and Mostafa Shabani

Correlation Analysis of Applications' Features: A Case Study on Google Play 202
A. Mohammad Ebrahimi, M. Saber Gholami, Saeedeh Momtazi, M. R. Meybodi, and A. Abdollahzadeh Barforoush

Information Verification Enhancement Using Entailment Methods 217
Arefeh Yavary, Hedieh Sajedi, and Mohammad Saniee Abadeh

A Clustering Based Approximate Algorithm for Mining Frequent Itemsets 226
Seyed Mohsen Fatemi, Seyed Mohsen Hosseini, Ali Kamandi, and Mahmood Shabankhah

Next Frame Prediction Using Flow Fields 238
Roghayeh Pazoki and Parvin Razzaghi

Using Augmented Genetic Algorithm for Search-Based Software Testing 248
Zahir Hasheminasab, Zaniar Sharifi, Khabat Soltanian, and Mohsen Afsharchi

Building and Exploiting Lexical Databases for Morphological Parsing 258
Petra Steiner and Reinhard Rapp

A Novel Topological Descriptor for ASL 274
Narges Mirehi, Maryam Tahmasbi, and Alireza Tavakoli Targhi

mmeybodi@aut.ac.ir

Pairwise Conditional Random Fields for Protein Function Prediction 290
Omid Abbaszadeh and Ali Reza Khanteymoori

Adversarial Samples for Improving Performance of Software Defect Prediction Models 299
Z. Eivazpour and Mohammad Reza Keyvanpour

A Systematic Literature Review on Blockchain-Based Solutions for IoT Security 311
Ala Ekramifard, Haleh Amintoosi, and Amin Hosseini Seno

An Intelligent Safety System for Human-Centered Semi-autonomous Vehicles 322
Hadi Abdi Khojasteh, Alireza Abbas Alipour, Ebrahim Ansari, and Parvin Razzaghi

Author Index 337

mmeybodi@aut.ac.ir

Efficient Cluster Head Selection Using the Non-linear Programming Method for Wireless Sensor Networks

Maryam Afshoon[1], Amin Keshavazi[1(✉)], Tajedin Darikvand[2], and Mahdi Bohlouli[3]

[1] Department of Computer Engineering, Marvdasht branch, Islamic Azad University, Marvdasht, Iran
Keshavarzi@miau.ac.ir
[2] Department of Mathematic, Marvdasht branch, Islamic Azad University, Marvdasht, Iran
[3] Department of Computer Science, Institute for Advanced Studies in Basic Sciences, Zanjan, Iran

Abstract. Wireless sensor networks consist of thousands of sensor nodes that are battery-based and have a limited lifetime. Accordingly, performance and energy efficiency are big challenges in wireless sensor networks. In this regard, numerous techniques were studied and developed to reduce energy consumption. In this paper, a mathematical-based method has been proposed for the optimal selecting of the cluster head in wireless sensor networks. In the proposed algorithm, a node was selected as a cluster head that has the maximum energy, weight, and density, as well as the lowest total distance from the other nodes. In this respect, the problem was converted into a math function, which was solved by non-linear programming. The experiment results show that the presented algorithm is efficient, as compared with the other approaches that have, hitherto, been used to solve this problem.

Keywords: Wireless sensor networks · Mathematical modeling · Non-linear programming · Clustering · Cluster head selection

1 Introduction

Wireless sensor networks are a combination of hundreds or thousands of battery-based sensor nodes and are a subset of the distributed systems. The battery of these nodes is limited and non-chargeable. One of the traditional methods for the energy efficiency of data transfer in these nodes is clustering [1]. The clustering process divides a geographical area into smaller parts and allocates a node as a head cluster to each part. Selecting a head cluster, which changes in each round, plays a critical role in data transfer energy efficiency [2]. The number of head clusters and the number of member nodes in each cluster can be constant or variable in the network. Also, nodes can directly send their data to the base station [3]. Wireless sensor networks have big challenges, including routing [3] and topology control [4]. The most important challenge of wireless sensor networks is energy efficiency [9]. Clustering is one of the most

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 1–12, 2020.
https://doi.org/10.1007/978-3-030-37309-2_1

mmeybodi@aut.ac.ir

important acts to handle this issue. Sensor nodes collect information from the surroundings and send it to the base station. Now, should all of the nodes send data simultaneously, problems like congestion, band width loss and error increase will occur, leading to an energy loss and therefore decreasing the network lifetime. To prevent these problems, for each cluster, a node must be selected as a head cluster node whose sensor nodes can separately send the collected information from the surroundings to their head cluster node and the head cluster node can send this information to the base station after collection and compression. The existence of the head cluster node is to decrease the number of connections in the network, leading to a reduction in the energy consumption and an increase in the network lifetime. The energy or battery lifetime, the sum of distances and the density are the key factors in head cluster selection. The algorithm chosen for clustering and head cluster selection is so effective in network energy consumption. In this paper, we have attempted to present a new method that works for clustering and head cluster selection based on mathematical optimization. The results of the simulation at the end of the paper show that this method has better outputs than the former methods in the same field; in the other words, by using this method, the networks lifetime will increase incredibly in comparison with the other methods.

Section 2 presents a literature review. Section 3 illustrates the non-linear programming method for cluster head selection in wireless sensor networks. Section 4 provides the simulation configurations. Section 5 depicts the evaluation results, and Section 6 concludes the paper.

2 Related Works

As it was said in the previous section, various algorithms are introduced for cluster head selection. The most famous algorithms in this field are:

The HEED [5] is a distributed protocol that selects the clusters independently of how the nodes are distributed based on the main parameter of the remaining value. In this protocol, the second parameter consisting of the node degree or proximity neighbors selection is also used. The HEED protocol selects the cluster heads according to the hybrid of node residual energy and a secondary parameter such as node proximity to its neighbors or the node degree. Moreover, HEED can asymptotically guarantee the connectivity of clustered networks [5].

The PEGASIS[1] [6] is a near-optimal chain-based protocol which is an improvement on the LEACH method. In PEGASIS [6], each node communicates only with a close neighbors and turns to the base station. Therefore, it reduces the amount of energy spent per round.

In [3], a new hybrid Genetic Algorithm (GA) and a K-means clustering namely EAR[2] has been proposed to maximize the network lifetime efficiency. This method

[1] Power-efficient gathering in sensor information.

[2] Energy Aware Routing.

mmeybodi@aut.ac.ir

uses the improved GA and the dynamic clustering network environment that is generated by k-means algorithm [3].

The new hybrid method of GA and fuzzy logic has been applied to balance the energy consumption among the CHs. In this method, the fitness function is calculated based on the difference of the current energy and the previous one. The BS selects the chromosome that has the minimum difference.

The fitness function is:

$$F = \left| E^{k}_{network} - E^{k-1}_{network} \right|$$

In the above formula, $E^{k}_{Network}$ represents the Energy in the round k (Energy flow in the network) and $Ek^{-1}_{Network}$ depicts the Energy of round k-1.

The algorithm contains a number of steps, which have been summarized here:

The first case is initialize the network (specifying the number of sensors). In the second phase, each node sends its position to its neighbors. Then, to calculate the "probability", fuzzy parameters such as energy, density, and centrality have been measured.

The nodes with a higher probability of fuzzy parameters will be selected as a candidate for cluster head. After that, GA is applied to select the cluster head. The cluster heads are presented to all nodes. Each sensor node joins to the nearest or adjacent cluster head and sends the information to the cluster head. The data aggregation is performed in each cluster head, and then the cluster head sends the received information of the package [3].

LEACH[3] [11], which is a self-organized clustering protocol, distributes the energy load to the network sensors. In this algorithm, the nodes organize themselves in the local clusters, therefore a node can act as a cluster in the cluster. High-energy nodes are randomly rotated to avoid the clogging the energy of the entire cluster network. Additionally, the data is locally aggregated to reduce the power consumption and increase the network life [11].

In this method [11], the nodes select themselves as a cluster head with a certain probability. These cluster heads inform the rest of nodes about their status. Each node chooses a cluster based on the minimum communication energy and becomes a member of selected cluster. When all nodes are organized into the clusters, each cluster head creates a scheduler for its nodes. Based on this scheduler, to saves the energy, the non-Cluster head nodes only turn on their radio when it comes to sending them, and in the rest of the time they are silent.

When the cluster node collects all members' data, it aggregates and compress the data and sends it to the base station. In this method, the nodes decide on their remaining energy. Each node decides independently of the other nodes. Therefore, additional negotiations are needed to diagnose the cluster head. LEACH is a cluster-based routing protocol in wireless sensor networks which is introduced in 2000 by Heinzelman et al. [11]. The purpose of this protocol is to reduce the energy consumption of nodes and improve the lifespan of the wireless sensor network.

[3] Low Energy Adaptive Clustering Hierarchy.

mmeybodi@aut.ac.ir

BCEE[4], which has been studied in [17], is a routing protocol that try to reduce energy consumption by balanced clustering of network nodes. In addition, more methods are designed for this purpose and are used in some cases. Some of them focus solely on head cluster selection like the evolutionary algorithms, data mining and fuzzy system.

In [7–9] the genetic algorithm and in [10] the ants colony algorithm and decision tree have been used for cluster head selection. The genetic algorithm is one of the best methods for determining the optimal points. In terms of input parameters and application of a set of functions and operators, one can propose a variety of methods based on the genetic algorithm for a single problem. Therefore, different researchers have presented various methods in this regard. Also, the genetic algorithm is one of the most famous and widely used evolutionary algorithms. It begins its work algorithm with a population of candidate answers (called chromosomes). During the implementation of this algorithm, the generation of chromosomes will gradually be improved and the subsequent generations will be generated in order to eventually satisfy the termination condition of the algorithm. In [12] the author has suggested a combined routing algorithm to develop the lifetime of network (Table 1).

Table 1. A review of different clustering algorithms in wireless sensor networks

Algorithm	Ref	Distribution	Selection Method	Stability	Energy efficiency
LEACH	[11]	Non-uniform	Random	Low	Low
TEEN	[16]	Non-uniform	Probable	Mid	Mid
Bayes	[14]	Non-uniform	Probable	High	High
PSO	[13]	Non-uniform	Probable	High	High
HSA	[13]	Non-uniform	Random	High	Mid
BCEE	[17]	Uniform	Random	Mid	Low
HSA-PSO	[13]	Non-uniform	Probable	High	High
Non-linear programming	This paper	Non-uniform	Probable	High	Very high

3 Proposed Algorithm

The method used in this paper is optimization with the non-linear modeling so as to choose an appropriate cluster head. The algorithm methodology has been depicted in Fig. 1.

Optimization in its own concept can be used to solve every engineering problem. The mathematical designing of a module is the main part in the mathematical optimization process. To obtain good relation results in achieving a proper optimized answer, a decision-making factor in a module should be introduced as a math function. The foregoing factor is called “target function”. There are various factors that affect the

[4] Balanced-clustering Energy Efficient.

mmeybodi@aut.ac.ir

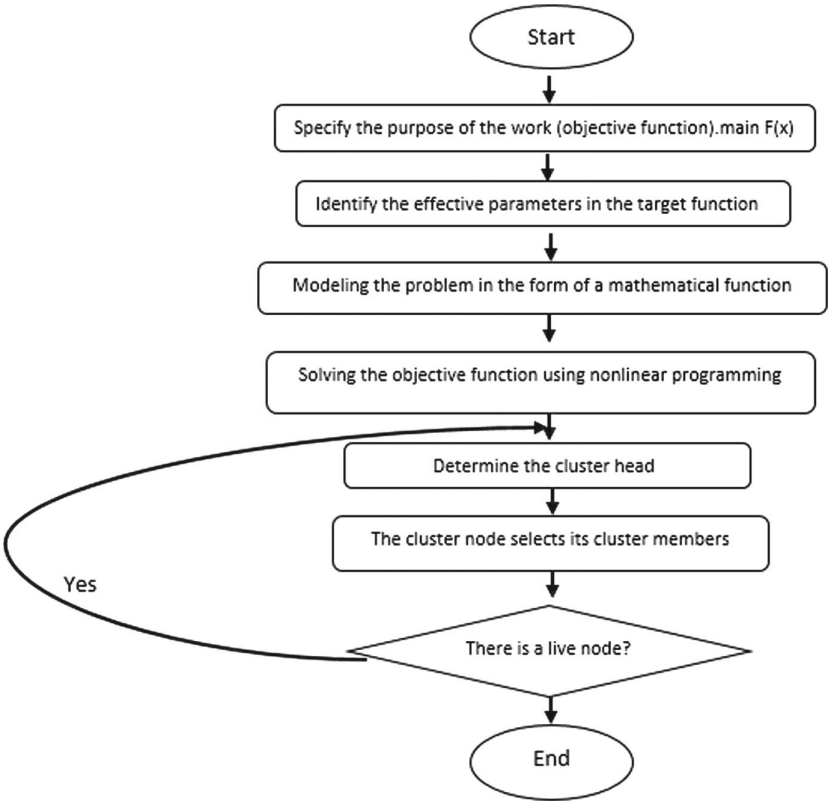


Fig. 1. Flowchart of proposed method

target function of a module and change its amount. These factors are introduced as parameters in the math pattern and are called "design parameters". In fact, the target function is written based on the foregoing parameters. The design parameters and target function are the two non-removable elements of each optimization problem. In the math designing of an optimization problem, the limitations are written as equality or inequality relations in accordance with the design parameters. It is noteworthy that some of the optimization problems have no limitations. In optimization problems, and among all of the accepted modules, the module that minimizes or maximizes the target function is called "optimized module" (according to the point that the problem will be minimized or maximized).

After recognizing all of the properties and parameters of a problem, we will write an appropriate math relation for optimization. In this mathematical pattern, the target function is a criterion for making decisions. The decision-making criterion with a combination of existing limitations will create a module. Writing the mathematical pattern of a problem is the most important part of optimization. The mathematical pattern can be written identically in all of the science and fields. This general module that consists of the target function, equality limitations and inequality limitations is as follows:

mmeybodi@aut.ac.ir

$$\begin{cases} Min\, F(x) & \{x\} \epsilon R^n \\ G_i(x) \leq 0 & i = 1 \ldots m \\ H_i(x) = 0 & j = 1 \ldots p \end{cases} \tag{1}$$

The function F(x) shows the target function of the problem that has to be minimized. Numbers n, m and p are the number of design parameters, inequality limitations and equality limitations, respectively [13].

After recognizing the necessities, limits and required criteria, a suitable math pattern is suggested and then solved.

Our point is to increase the lifetime of wireless sensor networks by selecting the proper head cluster node. The following factors are those which affect our point which increases the network lifetime; in the meantime, they are the parameters of the problem:

- Sum of distance
- Residual energy
- Density of nodes
- Weight of nodes
- Amount of initial energy of nodes

The target function of this module is considered as follows:

$$\text{Max(ED)} = (\text{Weight} * \text{density} * \text{Energy})/(\text{meandistance}) \tag{2}$$

And the limitation or the problem condition is:

$$\rho(i) = \sum_{i=0}^{n} \left(sumd(i) < \frac{d_0}{8} \right) \tag{3}$$

The method used in this paper utilizes a mathematic method based on non-linear modeling for the purpose of selecting the right head cluster node. In this method, at first we calculate the target function for each node and the head cluster node is the one with the maximum value of the target function. Then head cluster nodes begin to select their own members of the cluster (according to the density parameter or compression around the node) and in fact, they choose their own domain.

Distance between the nodes is calculated by the Euclidian relation (Eq. 4):

$$d = \sqrt{(x1 - x2)^2 + (y2 - y1)^2} \tag{4}$$

And in each round, the weight is changed based on a criterion or a condition. The foregoing parameter is renewed in each round of amounting according to the following relation that has two cases:

1. If the number of nodes that are selected as the cluster head node is less than 3, this parameter will be valued by the following relation:

mmeybodi@aut.ac.ir

$$Weight = weight - 0.5 \tag{5}$$

2. If the number of nodes that are selected as the cluster head node is more than 3, this parameter will be valued by the following relation:

$$Weight = weight - 0.25 \tag{6}$$

As it was said before, a module generally has several practical answers. Among all of the available answers, the best one is chosen, which is called "optimized answer". It is noteworthy that "optimization" is, in fact, a way to obtain the best answer.

In optimization problems, all of the relations, including the target function and limitations are introduced as the first power of the parameters. This is called "linear programming". Should at least one of the parameters enter the problem limitations or the target function with power more than one or in a form of non-linear functions like trigonometric functions or exponential functions, a non-linear optimization problem will emerge.

After making the relations of optimization problems, we have to solve them. There is not a unique method for the efficient solution to optimization problems. Because of this, various methods of optimization have been developed to solve different types of optimization problems. According to [13], the methods of non-linear optimization problem-solving are various, depending on the problem as to whether it is bound or unbound and whether it is linear or non-linear. The search method, repetitive method, binary gradient method, Newton method, modified Newton method, semi-Newton method, gradient picture method and multi-target optimization are some of the solving methods.

This paper has used a combined solving method, which is a combination of the repetitive method and convertor method (a change to the parameter). There is a loop in the repetitive part in each round, which calculates the target function for each node, and in the parameter changing part, we have used a weight parameter changing and renewing the parameter.

4 Simulation

We used Matlab software for simulation. We compared the input of our suggested method with the output of two other papers, and the result of this comparison is as follows:

In the beginning of the network, the nodes are randomly scattered in the environment and the environment dimensions have been shown in Fig. 2 (200 * 200). Also, the initial energy of the nodes at the beginning is considered as 2 J. Nodes with the shortest distance and the highest energy and density are chosen as a head cluster.

The consumed energy for the data transmission of the head cluster node must be calculated in each round, and based on that, the level of residual energy needs to be updated. According to (7), the amount of consumed energy for each member is calculated by the following relation:

mmeybodi@aut.ac.ir

The amount of lost or consumed energy for the head cluster node is calculated as follows:

$$E_{dis} = E_{Rx(l)}^{elec} + E_{Tx(l)}^{elec} + E_{DA} \tag{7}$$

The consumed energy to receive the cluster head node is calculated by the following relation:

$$E_{RX(l)} = l * E_{Rx}^{elec} \tag{8}$$

The amount of residual energy for each cluster head node at the end of each round:

$$E_{res} = E - E_{dis} \tag{9}$$

And for member nodes:

$$E_{res} = E - E_r \tag{10}$$

This procedure will be continued until the end of the network lifetime (until the death of the last node).

4.1 The Default Values and Assumptions

As it was said in the previous part, we have compared our input with the [14] article. In this comparison, we have used data from this paper. The constant data (the simulation parameters) of the [14] is as follows (Table 2):

Table 2. Default values

Parameter	Value
Number of wireless nodes	100
Operation environment	(200 * 200)
Location of the sync node	(100, 100)
Number of rounds	8000
Cluster head	30%, 5
E_{Tx}^{elec}	50 nj
E_{Rx}^{elec}	50 nj
ε_{fs}	10 nj
ε_{mp}	0.0013 pj
L	4000 bits
E_{DA}	5 nj

The amount of d_0 is calculated according to the following expression:

mmeybodi@aut.ac.ir

$$d_0 = \sqrt{(\varepsilon_{fs}/\varepsilon_{mp})} \tag{11}$$

and the initial energy of the nodes is joules.

5 Evaluation and Data Analysis

After running the proposed algorithm in Matlab software, and based on the default values from the previous part, the results of outputs have been shown in Figs. 2, 3, 4 and 5. Figures exhibit the number of existing living nodes in each round and the amount of the residual energy. It is completely evident that the suggested method has better results than the compared algorithms.

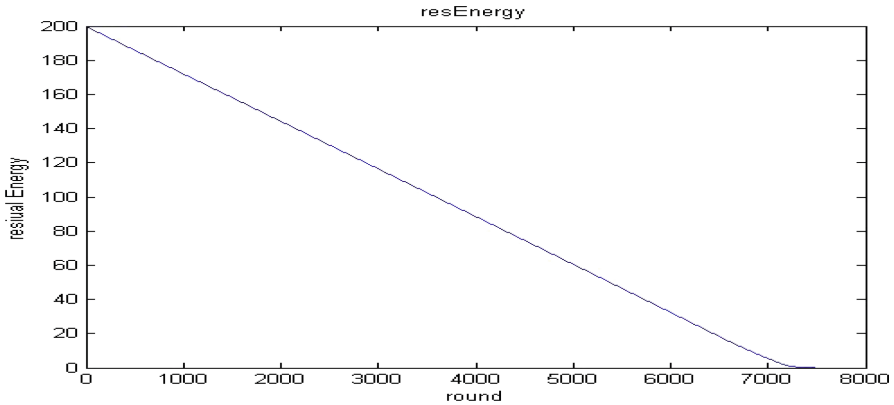


Fig. 2. The residual energy in each round

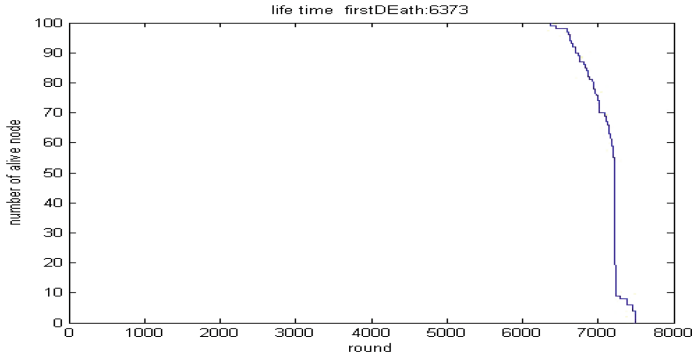


Fig. 3. The number of live nodes in each round

mmeybodi@aut.ac.ir

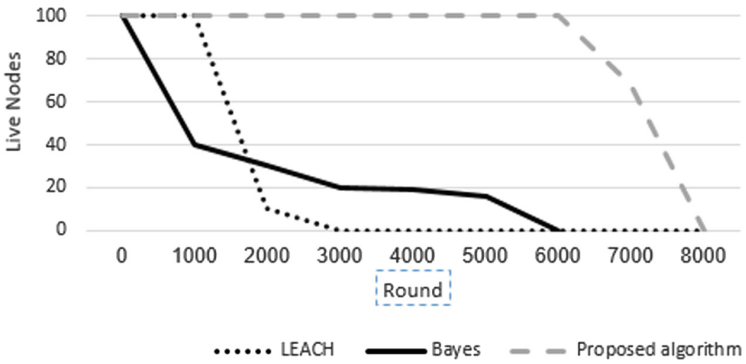


Fig. 4. The number of live nodes in each round, as compared with the other methods

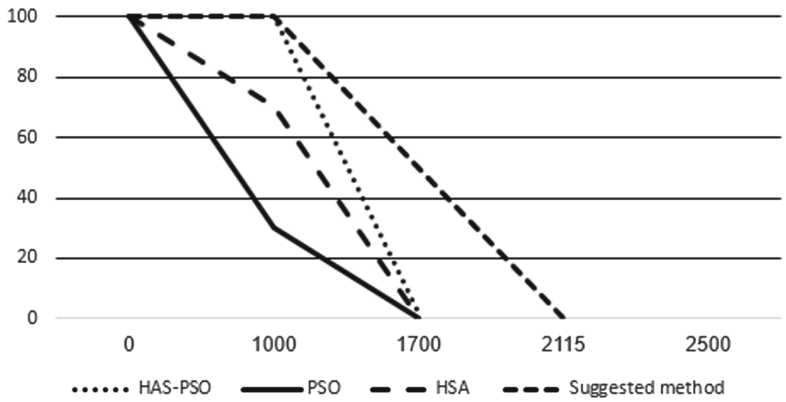


Fig. 5. The residual energy in each round compared with the HSA, PSO method and combined HSA-PSO method

Figures 4 and 5 are the results of the comparison of the proposed algorithm with the LEACH [11], Bayes algorithms [14], each of the HSA and PSO methods and their combination [13]. The comparison of the proposed algorithm with the other studied algorithms in this paper has been shown in Table 3. It is obvious that our proposed algorithm has the best performance in the optimization of the cluster head selection in wireless sensor networks.

Table 3. The results of the comparison of the proposed method with the related works

Algorithm	First dead node	Last dead node
Loss function according to Bayes [14]	95	6800
Leach algorithm [11]	1400	3500
PSO [13]	11	1600
HAS [13]	8	1680
Hybrid algorithm of HSA-PSO [13]	1304	1744
Suggested method	1181	2115

mmeybodi@aut.ac.ir

6 Conclusion and Future Works

As it was said before, mathematical optimization methods in choosing the head cluster in wireless sensor networks have rarely been used. Whereas using methods with a math basis have numerous advantages, including the algorithm flexibility. According to the results of the comparison of our proposed methods with Leach algorithm and loss function based on Bayes, we can realize that in our proposed method, further nodes could survive longer and the network lifetime is longer. This method will distribute the energy in the network in a completely equal and balanced manner and all nodes will survive until the last round, and in the final rounds all nodes start to lose their energy simultaneously, which is the best advantage of this method. It is evident that the proposed method has the best performance in the optimization of wireless sensor networks.

As it was said in the previous part, the suggested algorithm has high flexibility. Should someone be interested in working in this field, he can easily work and carry out research in this major by adding or removing parameters or by changing the available parameters. In addition, there are still many other methods to solve this problem. Interested researchers can use the following methods to select a proper head cluster like the honeybee method, intercross method and firefly (glow worm) method. Besides, they can utilize the linear or non-linear methods or a combination of these methods to reach better results.

References

1. Deosarkar, B.P., Yadav, N.S., Yadav, R.P.: Clusterhead selection in clustering algorithms for wireless sensor networks: a survey. In: 2008 International Conference on Computing, Communication and Networking, pp. 1–8 (2008)
2. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput. Surv. CSUR **35**(3), 268–308 (2003)
3. Amgoth, T., Jana, P.K.: Energy-aware routing algorithm for wireless sensor networks. Comput. Electr. Eng. **41**, 357–367 (2015)
4. Li, M., Yang, B.: A survey on topology issues in wireless sensor network. In: ICWN, p. 503 (2006)
5. Younis, O., Fahmy, S.: HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans. Mob. Comput. **4**, 366–379 (2004)
6. Lindesy, S., Raghavendra, C.: PEGASIS: power-efficient gathering in sensor information system. In: Proceedings of 2002 IEEE Aerospace Conference, pp. 1–6 (2002)
7. Barekatain, B., Dehghani, S., Pourzaferani, M.: An energy-aware routing protocol for wireless sensor networks based on new combination of genetic algorithm & k-means. Proc. Comput. Sci. **72**, 552–560 (2015)
8. Pal, V., Singh, G., Yadav, R.P.: Cluster head selection optimization based on genetic algorithm to prolong lifetime of wireless sensor networks. Proc. Comput. Sci. **57**, 1417–1423 (2015)
9. Hamidouche, R., Aliouat, Z., Gueroui, A.: Low energy-efficient clustering and routing based on genetic algorithm in WSNs. In: International Conference on Mobile, Secure, and Programmable Networking, pp. 143–156 (2018)

mmeybodi@aut.ac.ir

10. Kaur, S., Mahajan, R.: ACCGP: enhanced ant colony optimization, clustering and compressive sensing based energy efficient protocol (2017)
11. Cui, X.: Research and improvement of LEACH protocol in wireless sensor networks. In: 2007 International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, pp. 251–254 (2007)
12. Rao, S.S.: Engineering Optimization: Theory and Practice. Wiley (2009)
13. Shankar, T., Shanmugavel, S., Rajesh, A.: Hybrid HSA and PSO algorithm for energy efficient cluster head selection in wireless sensor networks. Swarm Evol. Comput. **30**, 1–10 (2016)
14. Jafarizadeh, V., Keshavarzi, A., Derikvand, T.: Efficient cluster head selection using Naïve Bayes classifier for wireless sensor networks. Wirel. Netw. **23**(3), 779–785 (2017)
15. Lloret, J., Shu, L., Gilaberte, R.L., Chen, M.: User-oriented and service-oriented spontaneous ad hoc and sensor wireless networks. Ad Hoc Sens. Wirel. Netw. **14**(1–2), 1–8 (2012)
16. Manjeshwar, A., Agrawal, D.P.: TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In: Null, p. 30189a (2001)
17. Cui, X., Liu, Z.: BCEE: a balanced-clustering, energy-efficient hierarchical routing protocol in wireless sensor networks. In: 2009 IEEE International Conference on Network Infrastructure and Digital Content, pp. 26–30 (2009)

mmeybodi@aut.ac.ir

A Survey on Measurement Metrics for Shape Matching Based on Similarity, Scaling and Spatial Distance

Bahram Sadeghi Bigham[1(✉)] and Samaneh Mazaheri[2]

[1] Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences, Zanjan, Iran
sadeghi@iasbs.ac.ir

[2] Faculty of Business and Information Technology, Ontario Tech University, Ontario, Canada
Samaneh.Mazaheri@uoit.ca

Abstract. Measuring difference or similarity between data is one of the most fundamental steps in data science. This topic is of the utmost importance in many artificial intelligent systems, machine learning and any data mining and knowledge extraction. There are several applications in image processing, map analysis, self-driving cars, GIS, etc., when data are in the shape of polygons or chains. In the present study, principal metrics for comparing geometric data are studied. In each metric, one, two or three features out of similarity, scaling, and spatial distance is considered. Evaluation for metrics based on three perspectives is discussed and results are provided in a detailed table. Additionally, for each case, one practical application is presented.

Keywords: Shape matching · Similarity · Measurement metric · Clustering · Data science

1 Introduction

Data science is one of the hot topics nowadays which is the knowledge of managing the existing data and extracting useful information to utilize in different situations. Some other topics such as data mining, big data, and data extraction also have the same objective. Comparison between data is a significant part of these topics. For instance, clustering and classification are not feasible without comparing data and computing the respected difference. In addition, there is a need for comparing data in all database queries.

To compare each type of data, there are special metrics. In this study, geometric data are on focused. Data are in the shape of polygons, path, tree, parts of a map, or simple shapes. There are three parameters in consideration, when comparing shapes; similarity (called first feature in this paper), scaling (called second feature), and spatial distance (third feature). At the time of evaluating similarity, scale of two shapes is not important; so the scaling changes in a way that two shapes have the most similarity.

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 13–23, 2020.
https://doi.org/10.1007/978-3-030-37309-2_2

mmeybodi@aut.ac.ir

Additionally, it is assumed that two shapes are overlapped and spatial distance will not make them different.

In fact, scaling means magnitude or measure of two shapes, which can be expressed in different ways, such as perimeter or area or combination of those. Third feature, i.e. spatial distance indicates the distance between two shapes. For example, between introduced metrics in this study, turning function is examined first features, i.e. similarity; while Fréchet distance assessing all three features at the same time.

When two shapes are compared, based on the application, one, two or all three features can be considered.

In next section, important metrics for measuring similarity between geometric shapes will be introduced, and in Sect. 3, a few applications of geometric data comparison that require different features will be discussed. In Sect. 4, a table including a comparison for different metrics has been presented in terms of considering each feature, which will help researchers to find out the most suitable metric based on their applications, and objectives, by considering the metric's capabilities. Section 5 will conclude the paper, and suggest future works.

2 Some Common Metrics

For comparing any two geometric objects, an appropriate metric is required. Several metrics have been proposed for this specific problem. However, in this study, only those methods which ignores definition and color are considered. Furthermore, learning techniques, and utilizing neural networks also excluded in this paper.

For simplicity, assume two polygons, two chains, or two cuts from a map are being compered together. In some applications, all of these three cases can occur. For instance, trajectories are the most common objects that mentioned metrics are applied on. Trajectories can be two simple chains, two simple polygons, or a piece of urban map. In trajectory topic, time is another dimension of the data, which is ignored in this study. In the following, some of the recognized metrics which have been used for this problem, will be discussed.

Essentially, trajectory is allocated into cohesive groups according to their mutual similarities. An appropriate metric is necessary [1–3].

Euclidean Distance [4]: Euclidean distance requires that lengths of trajectories should be unified and the distances between the corresponding trajectories points should be summed up,

$$D(X.Y) = \frac{1}{N}\sum_{i=1}^{N}\left((x_{1i} - y_{1i})^2 + (x_{2i} - y_{2i})^2\right)^{1/2}$$

where x_{ji} and y_{ji} indicate the i^{th} point of trajectories X and Y in Cartesian coordinate. N is the total number of points. In [4], Euclidean distance is used to measure the contemporary instantiations of trajectories.

mmeybodi@aut.ac.ir

Hausdorff Distance [5]: Hausdorff distance measures the similarities by considering how close every point of one trajectory to some points of the other one, and it measures trajectories X and Y without unifying the lengths in [6, 7],

$$D(X.Y) = \max\{d(X.Y).d(Y.X)\}, \text{in which}$$
$$\begin{cases} d(X.Y) = max_{x\in X} min_{y\in Y}||x - y|| \\ d(Y.X) = max_{y\in Y} min_{x\in X}||y - x|| \end{cases}$$

Bhattacharyya Distance [8]: Consider two data sets where both are divided into N sets. According to the distributions, each set would have its own frequency that the probability of occurrence of all data in each set, sums up to 1.

Bhattacharyya coefficient formula (e.g. ρ) is:

$$\rho(P.P\prime) = \sum_{i=1}^{N} \sqrt{P(i)P\prime(i)}$$

Maximum value for Bhattacharyya coefficient happens when each and every rectangle (probably the sets) are same with value equal to 1 and at most difference, this value is 0 or converges to 0.

Now Bhattacharyya metric is defined based on Bhattacharyya coefficient as follows:

$$d(p.p\prime) = \sqrt{1 - \rho(p.p\prime)}$$

This formula which treats in contrast of Bhattacharyya coefficient; whenever there is more similarity, Bhattacharyya coefficient yields to 1 and Bhattacharyya distance yields to 0.

This formula is defined for Bhattacharyya metric in range of [0, 1], to overcome this issue, this metric can also be defined as $d = -ln(\rho)$. In this case, with more similarity, the Bhattacharyya coefficient equals to 1 and $ln(1) = 0$, so Bhattacharyya distance becomes 0; and if there is less similarity, Bhattacharyya coefficient equals to 0 and $ln(0)$ is equal to infinite negative, thus the Bhattacharyya distance is reported as positive. To concise, $0 \leq d < \infty$ is the Bhattacharyya distance between two datasets where they are distributed in unity ranges.

Fréchet Distance [9]: Fréchet distance measures similarity between two curves by taking into account location and time ordering. After obtaining the curve approximations of trajectories X and Y, their curves map unit interval into metric space S, and a re-parameterization is added to make sure t cannot be backtracked. Fréchet distance is defined as

$$D(X.Y) = inf_{\alpha.\beta}\, max_{t\in[0.1]}\{d(X(\alpha(t)).Y(\beta(t)))\},$$

Where d is distance function of S, α, β are continuous and non-decreasing re-parameterization.

mmeybodi@aut.ac.ir

Dynamic Time Warping (DTW) Distance [10, 11]: DTW is a sequence alignment method to find an optimal matching between two trajectories and measure the similarity without considering lengths and time ordering [10, 11].

$$W(X.Y) = min_f \frac{1}{n} \sum_{i=1}^{n} \left| x_i - y_{f(i)} \right|$$

where X has n points and Y has m points, all mappings $f : [1.n] \rightarrow [1.m]$ should satisfy the requirements that $f(1) = 1, f(n) = m$ and $f(i) \leq f(j)$, for all $1 \leq i \leq j \leq n$.

Turning Function [12]: Another common method to compare, is to use Turning Function. In this method, shape analogy is done in the same scale, thus the difference in sizes would not matter and this metric merely discusses the similarity in shapes' structures.

To this point, first a plot with x axis representing the length and y axis representing angle in radian is considered. Then starting from one side of the shape, its angle to horizon is measured and inserted on the plot. In the next step, the next side and its angle is inserted on the plot. This process continues until it reaches the starting point. Similar process is applied for the other shape.

To compute the differences of two shapes using the Turning Function, it is enough to find the area between these two plots. But it is important to consider that by changing the starting point in shapes, various results are obtained. To overcome this issue, separate plots should be illustrated for each starting point, the acceptable answer is the one showing the least difference.

Longest Common Subsequence (LCSS) Distance [1]: $LCSS_{\varepsilon.\delta}(X.Y)$ aims at finding the longest common subsequence in all sequences, and the length of the longest subsequence could be the similarity between two arbitrary chains with different lengths.

Some more other distance types are proposed to consider more properties such as angle distance [13], center distance and parallel distance, which are defined as:

$$d_{angle}(L_i.L_j) = \begin{cases} |L_j| \times \sin(\theta); & 0 \leq \theta \leq \frac{\pi}{2} \\ |L_j|; & \frac{\pi}{2} < \theta \leq \pi \end{cases}$$

where θ is the smaller intersecting angle between L_i and L_j.

$$d_{centre}(L_i.L_j) = ||centre(L_i) - centre(L_j)||,$$

where $centre(L_i)$ and $centre(L_j)$ are the centre points of lines L_i and L_j [14].

$$d_{parallel}(L_i.L_j) = \min\{l_1 - l_2\},$$

where l_1 is the Euclidean distances of p_s to s_i and l_2 is that of p_e to e_i. p_s and p_e are the projection points of s_j and e_j onto L_i respectively [15].

mmeybodi@aut.ac.ir

3 Some Applications

As discussed earlier, in any application, some of the features, similarity, scaling and spatial distance are considered. The main objective of this study is to categorize these features, and introducing appropriate metrics for each specific application. Three applications of shape comparison which have different requirements for the mentioned features are discussed in the following.

3.1 Fingerprint Matching [16]

As it can be seen in Fig. 1, to compare two fingerprints, computing similarity is essential. Also, it is required to consider the scale for these two fingerprints. However, it is not necessary to consider the spatial distance.

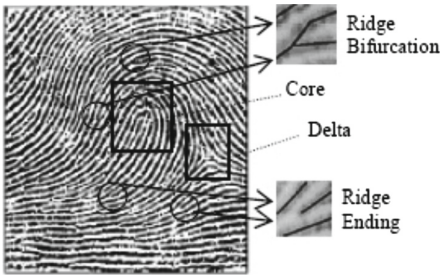


Fig. 1. Singular Points (Core & Delta) and Minutiae (ridge ending & ridge bifurcation) [16]

There are three different level of information contained in a fingerprint, namely, pattern (Level 1), minutiae points (Level 2), and pores and ridge contours (Level 3) [16].

There are a lot of contemporary fingerprint authentication systems which are automated and (Level 2 features) minutiae based. Minutiae-based systems rely on finding correspondences between the minutiae points present in "query" and "reference" fingerprint images generally. These types of systems normally perform well with high-quality images of fingerprints and enough surface area of fingerprint. However, these conditions are not always attainable [16].

In many situations, only a small portion of the "query" fingerprint can be compared with the "reference" fingerprint. This will lead to reduction of number of minutiae correspondences. In these cases, the matching algorithm would not be able to make a decision with high accuracy [16].

When the system is dealing with intrinsically poor quality fingerprints, only a subset of the minutiae can be extracted and used with enough reliability. Minutiae do not always constitute the best trade-off between accuracy and robustness, they may carry most of the fingerprint's discriminatory information. This fact has led the designers of fingerprint recognition techniques to look for other distinctive fingerprint features, other than minutiae which may be utilized in conjunction with minutiae

mmeybodi@aut.ac.ir

(consider that it is not as an alternative) to increase the system robustness and accuracy. It is worth mentioning that the presence of level three features provides detail for matching as well as potential for increased accuracy [16].

In this case, computing similarity and scale play an important role in fingerprint authentication systems, so the matching algorithm would be able to make a decision with high certainty [16].

3.2 Robot Pose Estimation

Autonomous navigation of a mobile robot is generally defined as control of robot motion to arrive at a given position in its environment without human intervention. This navigation task can be decomposed into the various aspects such as robot pose estimation (localization) and path planning.

To generate a path from an initial position to a target position, the localization is vital which frequently provides and updates a reliable estimate of the position and orientation of the robot given a global coordinate system in a structured or semi structured environment. Maintaining an estimate of the subsequent location of a robot using odometer, while the position and orientation of the robot is known at any time, is a common solution. But this method is sensitive to errors and the outcome is not reliable because of the integration of errors over time. Thus, the need arises for developing an algorithm to overcome the challenge.

One possible strategy to tackle the problem of pose estimation is to use scan matching, in which the geometric representation of the current scan is frequently compared to a reference scan until an optimal geometric overlap with the reference scan is obtained.

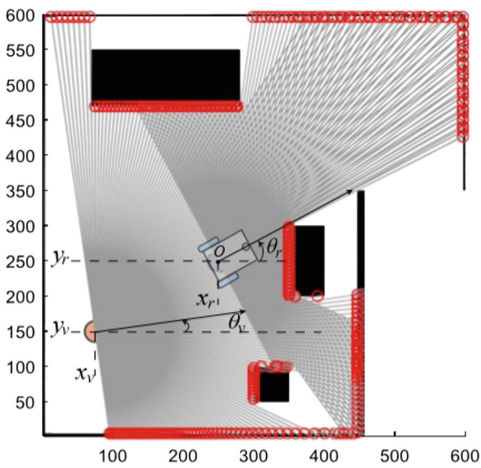


Fig. 2. RSD (real spatial description), and VSD (virtual spatial description) of a robot [18]

mmeybodi@aut.ac.ir

According to the reviews of literatures, scan matching, which is concerned with matching sensed data against map information, is an obvious choice in dealing with the self-localization problems. In efficient model-based approaches equipped with scan matching algorithms, by computing the best similarity between the robot observation, and an accurately-known or reconstructed map (i.e. the model) of the environment, it is possible to obtain a sensibly accurate estimate of the relative position and orientation of robot. In the proposed algorithm in [17, 18], firstly, a spatial description of the expected pose of the robot on a totally known or reconstructed environmental map is simulated, and then the simulated model is matched to the spatial description from laser range data.

They presented a new scan matching method (GSR) to yield a robust and fast pose estimation algorithm. The GSR algorithm takes two visualizations extracted from 2D laser range data, namely RSD (real spatial description), and VSD (virtual spatial description) (Fig. 2), and then tries to maximize the similarity between the two visualizations by transforming them (shift and rotate) into one coordinate system, and in this way, calculates actual difference between the two poses. VSD is a simulated visualization of the expected pose of the robot on a pre-calculated environmental map or, say, a simulation of sensor particles, and RSD is a visualization of the real pose of the robot from laser range data.

Robot pose estimation is one of the applications in which the selected metric needs to consider all three features. Since at the end, the shape from virtual vision should be the same as the shape from real vision of the robot; so all three features, similarity, scaling, and spatial distance must take into consideration. If a metric is selected for this application which do not consider scaling, robot can recognize a scene from two points, distant and close, the same scene, which will be resulted in a mistake recognition.

3.3 Polar Diagram Matching

The Polar Diagram [19] is a section of any planes with similar features to those of the Voronoi diagram. As a matter of fact, the polar diagram can be considered in the context of the generalized Voronoi diagram. Given a set S of n points on the plane, the locus of points having the smallest positive polar angle. The plane is sub-divided into various regions in a way that if the point $(x.y)$ lies in the area of s_i, it will be known that s_i is the first site found performing an angular scanning beginning from $(x.y)$. The boundary is the horizontal line crossing the top most site of S. An analogy between angular sweep and behavior of a radar can be drawn. In Fig. 3 can be seen that the polar diagram of an exemplary set of points on the plane.

To compare two polar diagrams to find out the probable modifications in regions related radars, there is a need for a metric that consider the spatial distance between two diagrams as well.

mmeybodi@aut.ac.ir

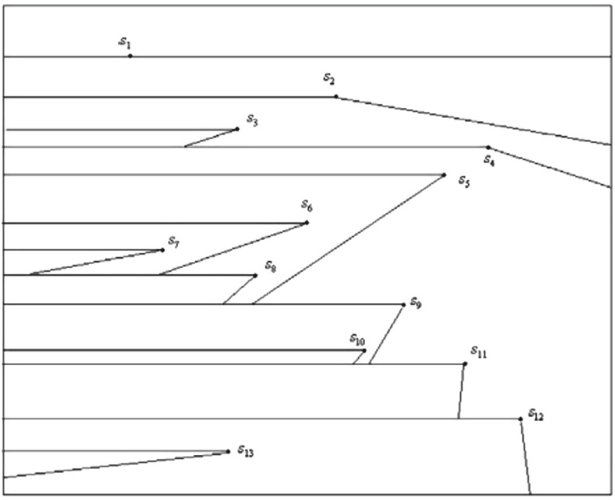


Fig. 3. Polar diagram of 13 sites [19]

4 Metrics' Properties

As discussed in detail, each metric considers some features to measure similarity between two geometric shapes. In Table 1, well-known metrics are presented with the related time complexity.

In last column, it is discussed that whether the metric requires to scale the geometric data before measuring distance between them or not. Based on the definition of metrics, Euclidian, Bhattacharyya, and turning function metrics are required to scale data before measuring.

In Euclidian metric, the number of points that are selected from each data should be equal. In Bhattacharyya metric, the number of intervals that data are distributed in, should be equal too.

Turning function scales data before calculating the difference between two geometric data. First, perimeter of two shapes set to 1, and then the similarity is computed. Other metrics do not require to scale data.

Third column is related to similarity; which metrics measure similarity between two data? At first glance, it should be mentioned that all the metrics measure the similarity, and if two shapes are not similar to one another, the result will show the difference between two shapes.

However, in a few metrics, like Hausdorff, this amount is so small, and in fact the biggest impact is related to difference in scale or spatial distance. The result from Hausdorff metric may show two different shapes; since the biggest impact is from spatial distance and the scale of the shapes (see Fig. 4). In Fréchet, Euclidian, and some of the metrics in the bottom of the table, the situation is the same.

mmeybodi@aut.ac.ir

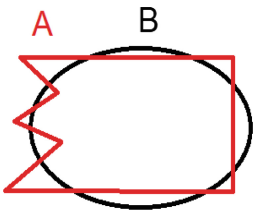


Fig. 4. Two different shapes which have small Hausdorff distance

Between mentioned metrics, turning function is a metric which explains this similarity feature and disregards scale difference and spatial distance between data. The situation is the same in Bhattacharyya metric, and in addition to that, data spatial distance will be considered as well without considering the volume of data.

Table 1. Metrics and their properties

Metric	Computational complexity	Similarity	Comparing the scale	Considering spatial distance between two shapes	Needs to scale scaling
Hausdorff distance	$O(mn)$	No	Yes	Yes	No
Fréchet distance	$O(mn)$	No	Yes	Yes	No
Euclidian distance	$O(n+m)$	No	No	Yes	Yes
Bhattacharyya distance	$O(m+n)$	Yes	No	Yes	Yes
Dynamic time warping distance	$O(mn)$	Yes	Yes	Yes	No
Turning function	$O(m+n)$	Yes	No	No	Yes
Longest common subsequence distance	$O(mn)$	Yes	No	No	No
Angle distance	$O(1)$	No	No	No	No
Center distance	$O(1)$	No	No	Yes	No
Parallel distance	$O(1)$	No	No	No	No

Forth column is related to another feature which volume of data, perimeter and area of data are also considered. Fréchet, Hausdorff and DTW are examples of the metrics that consider all three mentioned parameters. By using these metrics, when there is a greatest difference between magnitudes of two data, the difference after measuring would be higher.

As it is obvious from table's data, other metrics do not consider this feature when measuring. So if there is an application which needs to utilize this feature, it should be

mmeybodi@aut.ac.ir

modified first. For instance, LCSS metric does not include this feature in computation. However, it is possible to define the metric based on the length of largest common substring divide by largest given string, and in this way, the volume of given data will be considered in definition approximately.

Fifth column is related to spatial distance between two data. If there is a big difference between two data in terms of spatial distance, the question is, whether their difference would be a bigger number or not. In some applications like robot motion planning, geographical applications, and maps the answer is yes. However, in some applications like fingerprint matching this feature is not important; i.e. if two fingerprints are located far away from each other, it will not be related to spatial distance difference.

The most tangible case is seen in Fréchet metric; as is two polygons are located far away from each other, the length of the leash to control the dog should be longer. Reviewing this case in metrics such as Hausdorff, Euclidian, Bhattacharyya, DTW, and center distance is not complicated.

5 Conclusion and Future Work

Selecting the most suitable metric to measure the similarity between data in data science and data mining is of the importance. Several metrics have been introduced so far to compare two geometric data, which each metric has its own applications; i.e. each metric is appropriate to use in some special applications, and it is not suitable for other applications.

In some applications, only similarity between two geometric shapes is important, and difference in their magnitude as well as spatial distance is not affecting the similarity. However, sometimes it is required that in addition to similarity in appearance, the shapes will be the same in terms of scaling, and even spatial distance. In this study, multiple different applications as well as adverse metric are discussed for measuring the similarity between geometric data, and are evaluated based on three features including similarity, scaling, and spatial distance.

Results are presented in a table to provide an opportunity for researchers to select the most suitable metric for different applications. In future, applications of this table in data mining and also working with big data can be explored. Also, some metrics can be improved, so they can examine more features as well.

References

1. Morris, B., Trivedi, M.: Learning trajectory patterns by clustering: experimental studies and comparative evaluation. In: IEEE Conference on Computer Vision and Pattern Recognition 2009, CVPR 2009. vol. 9, pp. 312–319. IEEE (2009)
2. Zhang, Z., Huang, K., Tan, T.: Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 3, pp. 1135–1138. IEEE (2006)

mmeybodi@aut.ac.ir

3. Atev, S., Miller, G., Papanikolopoulos, N.P.: Clustering of vehicle trajectories. IEEE Trans. Intell. Transp. Syst. **11**(3), 647–657 (2010)
4. Nanni, M., Pedreschi, D.: Time-focused clustering of trajectories of moving objects. J. Intell. Inf. Syst. **27**(3), 267–289 (2006)
5. Borwein, J., Keener, L.: The Hausdorff metric and Cebysev centers. J. Approximation Theory **28**(4), 366–376 (1980)
6. Liu, M.-Y., Tuzel, O., Ramalingam, S., Chellappa, R.: Entropy-rate clustering: cluster analysis via maximizing a submodular function subject to a metroid constraint. IEEE Trans. Pattern Anal. Mach. Intell. **36**(1), 99–112 (2014)
7. Chen, J., Wang, R., Liu, L., Song, J.: Clustering of trajectories based on Hausdorff distance. In: 2011 International Conference on Electronics, Communications and Control (ICECC), pp. 1940–1944. IEEE (2011)
8. Li, X., Hu, W., Hu, W.: A coarse-to-fine strategy for vehicle motion trajectory clustering. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 1, pp. 591–594. IEEE (2006)
9. Dowson, D.C., Landau, B.V.: The Fréchet distance between multivariate normal distributions. J. Multivar. Anal. **12**(3), 450–455 (1982)
10. Shao, Z., Li, Y.: On integral invariants for effective 3-D motion trajectory matching and recognition. IEEE Trans. Cybern. **46**(2), 511–523 (2016)
11. Bautista, M.A., Hern´andez-Vela, A., Escalera, S., Igual, L., Pujol, O., Moya, J., Violant, V., Anguera, M.T.: A gesture recognition system for detecting behavioral patterns of ADHD. IEEE Trans. Cybern. **46**(1) 136–147 (2016)
12. Latecki, L.J., Lakamper, R.: Shape similarity measure based on correspondence of visual parts. IEEE Trans. Pattern Anal. Mach. Intell. **22**(10) 1185–1190 (2000)
13. Lee, J.-G., Han, J., Whang, K.-Y.: Trajectory clustering: a partition-and group framework. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, pp. 593–604. ACM (2007)
14. Lee, J.-G., Han, J., Li, X., Gonzalez, H.: TraClass: trajectory classification using hierarchical region-based and trajectory-based clustering. Proc. VLDB Endowment **1**(1), 1081–1094 (2008)
15. Li, Z., Lee, J.-G., Li, X., Han, J.: Incremental clustering for trajectories. In: International Conference on Database Systems for Advanced Applications, pp. 32–46. Springer (2010)
16. Mazaheri, S., Bigham, B.S., Tayebi, R.M.: Fingerprint matching using an onion layer algorithm of computational geometry based on level 3 features. In: International Conference on Digital Information and Communication Technology and Its Applications. Springer, Heidelberg (2011)
17. Shamsfakhr, F., Bigham, B.S.: A neural network approach to navigation of a mobile robot and obstacle avoidance in dynamic and unknown environments. Turk. J. Electr. Eng. Comput. Sci. **25**(3), 1629–1642 (2017)
18. Shamsfakhr, F., Bigham, B.S.: GSR: geometrical scan registration algorithm for robust and fast robot pose estimation. Assembly Autom. (2018, to be printed)
19. Sadeghi, B., Mohades, A., Ortega, L.: Dynamic polar diagram. Inf. Process. Lett. **109**(2), 142–146 (2008)

mmeybodi@aut.ac.ir

Check for updates

Static Signature-Based Malware Detection Using Opcode and Binary Information

Azadeh Jalilian[1(✉)], Zahra Narimani[1,2], and Ebrahim Ansari[1,2]

[1] Faculty of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences, Zanjan, Iran
jalilian.azadehl@gmail.com
[2] Research Center for Basic Sciences and Modern Technologies (RBST), Institute for Advanced Studies in Basic Sciences, Zanjan, Iran

Abstract. Internet continues to evolve and touches every aspect of our daily life thus communications through internet is becoming inevitable. Computer security has been hence becoming one of the important concerns of internet users. Malware, a malicious software, is a harmful code that poses security thread for infected machines, thus malware detection has become one of the most important research topics in computer security. Malware detection methods can be categorized into signature-based, and behavior-based methods; each of which can be performed in a dynamical or static behavior. In this paper, we describe a static signature-based malware detection method based on opcode and binary file signatures. The proposed method is based on N-gram distribution and is improved using a proposed Top K approach which suggests selecting top most similar k files in classification of a new unknown file. The results are evaluated on VXheaven malware binaries, and windows system files are used as a repository of benign binaries.

Keywords: Malware · Signature-based malware detection · Static analysis · Opcode

1 Introduction

Organizations and individual computers are being infected by malwares every day. Base on annual security report on 2014, malicious attacks increased 700% from 2012 to 2013. Over 552 million secret identities were in risk in 2013 which indicated a 493 percent rise in attacks in 2012. Most of these attacks involved some kind of a malware [1]. A large number of malwares has been identified so far, and on the other hand new malicious software is being produced thus urging the need for efficient malware detection strategies. There is a continuous competition between malware creators and malware preventers [2, 3]. Researchers working on malware detection, a subcategory of computer security, try to develop algorithms and methods to distinguish malicious files from benign files. Malware detectors are designed to identify malicious software, by detecting their malicious behavior. By identification of malicious software, the access of programs or users can be controlled based on their identity; this leads to a safe environment to run safe code [1].

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 24–35, 2020.
https://doi.org/10.1007/978-3-030-37309-2_3

mmeybodi@aut.ac.ir

The main approaches for malware detection include ones using behavior-based and ones using signature-based techniques. Besides, the mentioned analysis techniques may be performed using a static or dynamic analysis [4]. The idea behind behavior-based approach is to identify files with similar behavior, which can be used beside machine learning methods in order to provide a mechanism for classification of a malicious behavior. Behavior-based analysis depends on execution traces of files generated on an emulated environment, and can be helpful in detecting malware with different syntax but similar execution profile (behavior). Behavior-based methods are sensitive to false positive rate [5].

Signature-based approach is the common approach in most of antivirus tools. Features extracted from disassembled code of binary files, generated using disassemblers or debuggers, are used to create signatures. A malware family is identified by aforementioned features. In 2016, a number of new malware was approximately 127 million and for the first time in history, it was less than the year before (144 million). About 22 million new kinds of malware in the first quarter of 2017 shows that the number of malicious files is decreasing. However, this decrease is only in the number of new malware and malware attacks generally are increasing and this shows the importance of malware detection (Especially by using signature-based methods) [6].

Signature-based methods are faster and more secure than behavior-based methods for malware detection. In static analysis, the executable code is analyzed without actual execution; what is done is extraction of code's low-level information generated using disassembler tools. The advantage of static analysis is consideration of the whole code structure. This is in comparison to dynamic analysis, which only considers the behavior of malware observed from its execution. A simulated environment such as virtual machine, emulator, or sandbox can be used to execute the files. While static analysis suffers from its weakness in existence of code obfuscation, dynamic analysis may fail to detect a potential execution path, which has been not executed due to existence of many trigger conditions, representing malicious behavior [4]. Hiring anti-emulation and anti-virtual machine tools by malware developers, can also disrupt the functionality of dynamic analyzers [6].

Computer virus detection was introduced to the world of malware detection in 1983 by Cohen for the first time, who formalized the term "computer virus" [6]. Malware can be divided into various groups such as virus, worm, Trojan, spyware, adware, and a combination of these categories or a sub-group of them [5, 7].

The first malware to exist was called a virus, for the resemblance of its mechanism to the biologic virus [8]. A computer virus is a code which cannot do anything all by itself. This code should be injected into another program's code in order to be executed. This program is called an infected program. The other characteristic of a computer virus is that it can replicate itself and infect other programs after being executed [8]. When the systems are infected, this code fulfills its goal. Viruses can be designed to perform harmful operations on a computer such as spying, overthrowing, causing a disturbance in systems and following military goals, to mention but a few. Following viruses, stronger malware like worms and rootkits were created which have great abilities [9].

Old methods of signature-based detection and developed and new ones can detect this malware. To prevent detection by anti-viruses, malware writers use methods

mmeybodi@aut.ac.ir

known as obfuscating. Ease of implantation, speed, and security of signature-based methods made us use this method alongside extracting new features in this study. The focus of this study, for the most part, is on static malware analysis, but there are some parts of dynamic analysis methods. Eventually, the suggested approach will be introduced, examined, and explained.

2 Previous Work

In recent years, there has been so much concern about malware detection. Behavior-based and signature-based malware detection methods can both use static, dynamic or hybrid analysis methods (see Fig. 1).

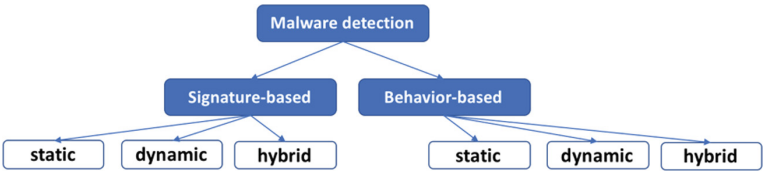


Fig. 1. A classification of malware detection techniques.

Static detectors investigate the complete malware code (independent of whether or not the code will be executed in practice) and hence can have a thorough structural analysis of the malicious code without running programs, whereas dynamic detectors run malware in an isolated environment as virtual machine and aim to analyze the behavior of the malicious code during its execution. Static analysis for malware detection therefore can be implemented using binary sequences, order execution sequence or opcodes. The same approach is used in the current research.

Primary research on malware detection focuses on analyzing structural features of binary codes [10, 11]. Considering malware detectors as binary classifiers, aiming to classify files into benign or malicious categories, binary files were used by Li et al. [10] to generate fingerprints to be used for classification purposes. Weber et al. tried to find some statistical homogeneity, such as patterns of instruction, entropy metrics, and jump/call distances, in benign files and discovered that malicious files are likely to break this homogeneity. Static and dynamic analysis of binary files were undertaken by several researchers afterwards [12–15]. Bilar for the first time proposed the use of opcode as an alternative for binary files [14]. An opcode is a part of machine instruction which determines the operation to be executed by machine. Particularly, a program is defined as a set of assembly instructions. Analyzing opcodes within a piece of code, can reveal patterns to classify that part of code into benign or malicious code. Bilar [16] illustrated the functionality of opcodes as features to be used in malware detectors through investigating the distributions of opcodes in malicious code. Some opcodes has been identified to be a better predictors, determining 12–63% of the variation, in their experiments on 67 executable file.

mmeybodi@aut.ac.ir

Santos et al., proposed various kinds of malware detection methods using procedures based on opcode sequence [17]. In their first work, a procedure focused on detecting various kinds of malware using repetition numbers of opcode sequence is offered, to create a show of executable files [17]. They used opcodes and the creative approach of N-gram sequence analysis for malware detection. To achieve this, they only used 1-gram sequence in the first step. After the results of these experiments came out, it became clear that 1-gram sequences are not enough for malware detection and they don't have the necessary information in order to achieve a powerful classifier. The splitting line between malicious and benign files was not suitable and detection was poor and unreliable. They understood that this sequence doesn't work well, so they applied the combination of 1-gram and 2-gram sequences.

Sung et al. [18] proposed a method named Static Analysis for Vicious Executables (SAVE). In their proposed method, the signature of a malware is showed as an API call. A 32-bit number represents each API call. The most significant 16 bits relate to the API calls, while the least significant 16 bits relate to the API functions position in a vector of API functions. The Euclidean distance between detected signatures and API sequences found in the target program is calculated. The average of three similarity functions determines the similarity of API sequence of the target program with the existing signatures in the database. Three similarity metrics were used in the experiments; cosine similarity, extended Jaccard measure, and Person correlation measure.

Shabtai et al. [19] use opcode sequence and N-gram procedure for malware detection in two phases. This approach is represented in two phases; training and test phase. In the training phase, a set of benign and malicious training files are represented to the system. Each file is converted to a feature vector, based on a set of specified opcodes. Feature vectors in training set is an input for learning algorithm such as Artificial Neural Network or Decision Tree. After processing these vectors by learning algorithm, a model for classification is proposed. A test set of new benign and malicious files, which are not represented to the model in the training phase, is classified by the model created in the training phase. Each file in the test phase is disassembled for the first time and its representative vector will be extracted. The trained model, categorizes the file as benign or malicious, based on this feature vector. In the test phase, the classifier's efficiency is assessed by standard and accurate measures of categorization. Thus, knowing the true family of files (data labels) is necessary in order to make it possible to compare the real family with the family predicted by predictor model in test phase.

3 Proposed Method

Signature-based malware detection with the static procedure is introduced formerly in previous work. This method is widely used in commercial anti-viruses. A signature-based method is faster than other approaches in malware detection. This method emphasizes on extracting file features in a static mode and without running them. One of the advantages of this method is the fact that the system will not be infected by malware. Also, its performance and accuracy is high for known malware families. Most of these method attempt to detect malware files (or families), by examining the

mmeybodi@aut.ac.ir

structures of executable files and extracting their features such as binary sequences, opcodes or API calls.

Our proposed method is based on a new approach of using combination of opcode features with different degrees. This approach is implemented and also tested in combination with different binary sequence features. Among thousands of malware in the computer world, number of those with unique execution pattern is probably very few [20]. It means that the majority of the executable code of each malware is the same as the rest of the malware probably from the same family. Thus, not a lot of unique execution is observed. Thus, finding strong predictors (features) can be the key for proposing a good classifier. The experiment results reported in results section confirms the strength of our proposed features.

Our method includes three phases; Extracting opcode and binary sequences from benign and malicious files, generating N-grams, classifying files into benign and malicious groups using a classification algorithm. Opcode sequences and binary will be examined by the N-gram approach. Each N-gram represents a feature. With larger values of N, the number of input features, equal to $\binom{k}{N}$, increases dramatically. On the other hand, smaller values of N, for example N = 1, does not have the ability to capture adequate information from structural characteristics of opcodes encoded by all possible opcode combinations. As a result, choosing the right value of N, leading to maintaining the performance and space efficiency at moderate level while not losing important information of the feature space, is critical in order to have an efficient and accurate malware detector. A solution for this problem is to choose a strategy that decreases the computation overhead while benefiting from optimized sequence features.

Another important component of the malware detector, is the classifier. Malwares tend to share traits, leading to categorization of them into families [21]. In case malware family information is available, the classifier can be designed based on specific family signatures [21, 22]. When no information about malware families is provided, the classification task should consider only the file labels regarding the file being malware or benign. In this case, the classification task can be more complex.

Since we assume here that we don't have any information about the malware families, the only feature available for supervised learning would be file status regarding being malware/benign. The classifier's task is now to predict the label of an unclassified sample with regard to the label of similar files in the training set. The similarity between the file to be classified and all the files in the training set can be measured using a criterion such as cosine similarity. In the prediction phase, we consider only Top-K similar files though, and label the new file by finding the dominant label within these Top-K similar files. This approach is similar to K nearest neighbors method for classification. In case malware family information is available, the prediction can be made based on similarity to malware families.

We used file binary information (instead of opcodes) within the same procedure and our observation was that using binary information within the same procedure also leads to acceptable results (reported in result section). Since we have to limit the set of input features (by keeping the value of N – in N-grams – moderate), we decided to

mmeybodi@aut.ac.ir

improve the feature space strength by adding features extracted from binary files to the features extracted from opcodes. Details is provided about preparation of training/test data, preprocessing and feature extraction, and finally classification. Finally, the experiments and results is provided.

4 File Selection

In this study, we used 32-bit Portable Executable (PE) files as our benign dataset. The PE+32 format is for 64-bit Windows, which has some differences to 32-bit PE. There are no new fields in PE+32 format. Most of the changes are to make the conversion of the field from 32-bit to 64-bit easier. The structure of PE file is demonstrated in Fig. 2. Some parts, such as troubleshooting information that is located at the end of the file, might be read but be absent from memory. PE header provides us information, such as how much of the memory to be assigned to run the intended program by the computer. In PE, the code section includes the code and the data section include various types of data, such as input and output tables of API, recourses, and relocations. Each of these parts has its own memory attributes [23].

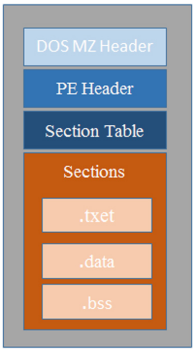


Fig. 2. PE file

In order to collect the benign file part of our dataset, we used system files from a malware-free Windows. We selected these files from drive C, folder "Program files (X86)", which has various programs such as compilers (Visual C, Visual C+, and Visual Basic) for Windows (32-bit and 64-bit in PE format), Internet browsers, pdf reader, paint, etc. The malicious files are downloaded from VXheaven computer virus collection [24], which has consists a set of different kinds of malicious files. The subset contacting 32-bit Windows malware is chosen as the malicious dataset. We analyzed different types of Viruses, Worms, Rootkits, etc. The purpose of this study was to detect malware without having their family information. Thus, we ignored the information regarding to malware families in our training phase. The size of the malware ranges from 2 kB to 2 MB, and the size of the benign files from 8 kB to 380 MB.

mmeybodi@aut.ac.ir

5 Preprocessing Files and Features Extraction

The files used for opcode and binary extraction should not be compressed, therefore the files are decompressed in the first step if necessary. After disassembling, the majority of files should contain opcodes. File disassembly can be performed using dynamic or static method. Dynamic dissembling is performed while the program is being executed. The main issue with this approach is that only a limited possible execution paths can be taken while execution, and some part of the code may remain unexecuted (for example because of conditional statements, such as malware code which are set to run on a specific date). On the other hand, in static analysis, the whole program is disassembled, leading to thorough extraction of structural features.

File dissembling was performed by PE explorer program and statically in this work. PE explorer receives the collected 32-bit executable files as input file and saves the assembly codes of these files, which include the intended opcodes, as output. Data preprocessing is time-consuming and requires high precision.

6 Feature Extraction, Similarity Measure, and Classification

We used N-gram technique to form the feature space generated from opcodes. The benefit of using N-gram is its simplicity, and its stability in presence of obfuscation. Owing to the fact that malware writers always try to prevent their malicious codes from being detected, they use obfuscation methods to achieve this goal. Using cosine similarity measure ignores the order of instructions and repetition of opcode or binary sequences, hence able to reveal malware similarities even though the code is obfuscated.

A file can be seen as a vector of features (N-grams of opcodes or binary sequences). Cosine similarity quantifies the similarity between two vectors, which are N-dimensional vectors corresponding to N-grams in this case. Each element of these vectors can be 1 or 0, regarding the presence or absence of the corresponding N-gram. Cosine similarity is defined as:

$$Cosine\ Similarity\ = \frac{v_k^{'} v_u^{'}}{\left\|v_k^{'}\right\|^2 * \left\|v_u^{'}\right\|^2} \tag{4}$$

where, $v_k^{'}$ and $v_u^{'}$ are the two vectors for which we want to measure the similarity. To decide whether an unknown file $v_u^{'}$, belongs to benign or malware category, its similarity to files with known type ($v_k^{'}$) is measured and the unknown file class is predicted using the class of Top-K most similar known vectors.

Measuring the similarity between vectors existing in our training data, we observe that the dispersion of the similarity rate of benign and malicious files of each vector is different. For instance, similarity rate of benign files in 3-gram vector is between 0.5 to 0.9, but similarity rate of malicious files in the same vector is between 0.1 to 0.4. To avoid this bias, we applied normalization. By normalization, the dispersion of the similarity measure of files will be the same and in the same range [0, 1].

mmeybodi@aut.ac.ir

7 Evaluation

The number of True/False Negative/Positives should be computed in order to measure the classification performance:

- True positive (TP): Number of malicious programs which are correctly categorized as malware.
- True negative (TN): Number of benign programs which are identified as benign files.
- False positive (FP): Number of benign files which are incorrectly categorized as malware.
- False negative (FN): Number of malicious files which are incorrectly categorized as benign files.

We use sensitivity, specificity, and accuracy (Eq. 5–7) for evaluating our final result.

$$Sensitivity = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

$$Specificity = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FN}} \tag{6}$$

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TN} + \mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{7}$$

8 Implementation and Results

In our first experiment, the similarities of 1-gram, 2-gram and 3-gram opcode sequences and reported them, the results of which can be seen in Table 1. In these experiments, the training set consists of 216 benign and 203 malicious files. In the first three following experiments, the nearest neighbor (using cosine similarity measure) is used to label an unknown sample.

Table 1. Results of the implementation of various degrees of opcodes to percentage

Type	1-gram	2-gram	1, 2-gram	1, 3-gram	2, 3-gram	1, 2, 3-gram
Sensitivity	3.45	94.58	3.94	99.51	99.51	98.52
Specificity	99.54	19.91	99.53	57.41	57.41	57.41
Accuracy	52.98	53.69	53.22	77.80	77.80	77.32

As it can be observed from our results in Table 1, 1-grams are not strong features for classification purposes. The reason is presence of similar opcodes such as mov, jz, pop, push, etc. that by themselves does not have any relevancy to the class of files, but

mmeybodi@aut.ac.ir

at the same time, they play a significant role in similarity between files (as 1-grams) since they are frequent in all the existing files in training set. Using 2-grams has improved the sensitivity (the ratio of correctly predicted malwares to all malwares available in the training set). Combinations of 1, 2 and 3-grams represents a feature set with highest classification strength.

The same experiment is repeated using binary sequences (results are provided in Table 2). Due to the results, binary sequences yield in good classification performance in detecting malicious files, but are extremely non-functional in detecting benign files, so the general accuracy of this method is very low.

Table 2. Implementation of various degrees of binary to percent

Type	2-gram	3-gram	2, 3-gram
Sensitivity	96.55	97.54	94.58
Specificity	15.74	13.89	19.91
Accuracy	54.89	54.41	56.08

Due to the high detection rate of malicious files by binary sequences, we decided to combine opcode and binary sequences to improve test results. In the next experiment, two binary vectors consisting of 2-gram, and 3-gram sequences, and three vectors of opcode sequences consisting of 1-gram, 2 g, and 3-gram sequences, are used as input features. Result is presented in Table 3; the sensitivity reached 100 percent, the specificity didn't change much, and the accuracy has increased.

Table 3. Results of implementation of combining binary and opcode sequences to percent

Type	All
Sensitivity	100
Specificity	57.41
Accuracy	78.04

To improve the results, we decided to use the Top-K idea. This idea is about examining the similarity of each file with K of most similar files to it. This criterion improves the classification efficiency since it prevents the noise effect and also excludes dissimilar files to has an effect on the prediction of the label of the file to be classified. Using Top-K approach, which decreases computational load and cancels noise (not examining dissimilar files), helped to increase accuracy in malware detection. Since the behavior and execution pattern of different families of a malware are not similar to each other (for example, a family deletes files, but another one replicates them), using the Top-K idea can increase detection accuracy, as it prevents the calculation and examination of similarity between two different families. The reason is that the similarity of files belonging to the same family is higher, and they automatically are put into the Top-K selected for prediction task.

mmeybodi@aut.ac.ir

Figure 3 shows the effect of various degrees of Top on 1, 2, 3-gram opcode sequences. The most similarity idea has significantly improved detection accuracy, and the highest accuracy belongs to Top-10, with 86.63%. Increasing K, will improve the efficiency until some threshold (threshold = 10 in our validation set), and afterwards leads to decrease of accuracy. The reason is inclusion of non-similar files for prediction.

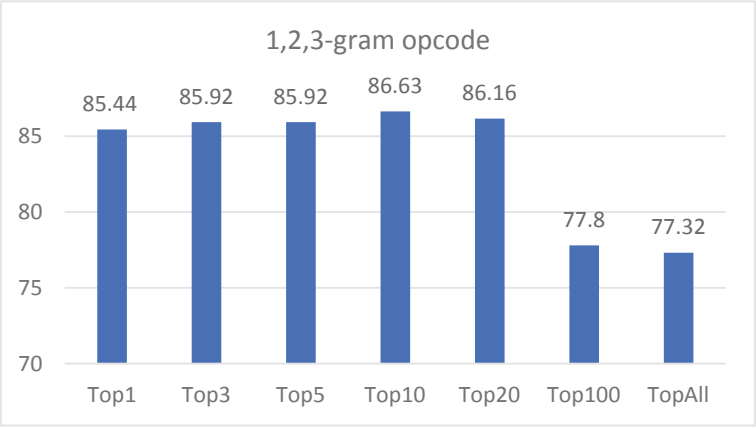


Fig. 3. Effect of various combinations of Top on 1, 2, 3-gram combinations of opcodes

Figure 4 shows the effect of various degrees of Top on 2, 3-gram binary sequences. The most similarity idea has significantly improved detection accuracy, and the highest accuracy belongs to Top-5, with 81.14%.

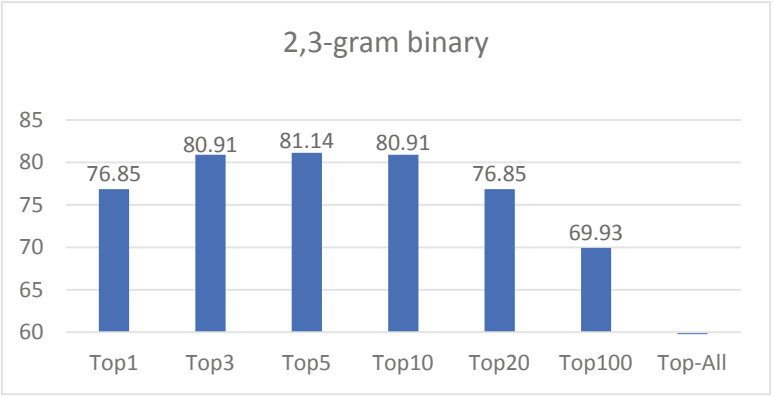


Fig. 4. Effect of various degrees of Top on 2, 3-gram binary sequence. While the Top-All score is 54.89, it hasn't been shown in the chart.

mmeybodi@aut.ac.ir

Figure 5 shows the effect of various degrees of Top on Combination of Opcodes and Binary sequences. In the combination, the K is selected to be 3, using a validation set, resulting in achieving accuracy of 86.39.

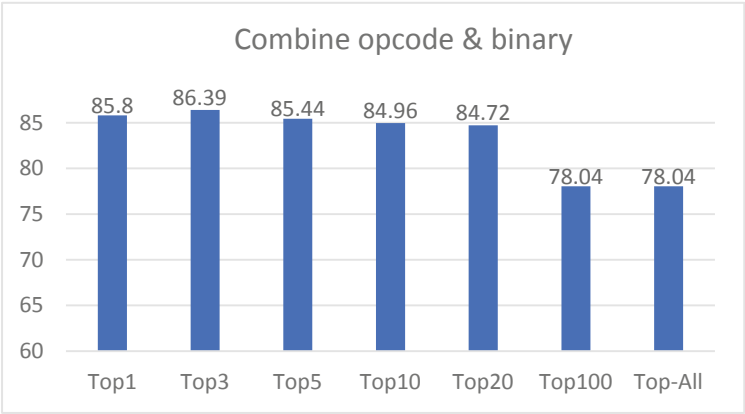


Fig. 5. Effect of various combinations of Top on combination of opcodes and binary

9 Conclusion

In this study, the combination of 1, 2-gram opcode sequences were evaluated for detecting malware files. The results were significantly better and more hopeful than 1-gram opcode sequence used previously. The combination of the 2, 3-gram and 1, 2, 3-gram sequences were implemented to detect existing malware. The performance of binary sequence (2-gram and 3-gram binary sequences, and their combination) were also experimented for malware detection purpose. The results of this examination were not as good as the results using opcode sequence features, but the classification was improved for the case of detecting benign files. Combination of binary and opcode sequence then was used for classification of malware/benign files. Together with the proposed Top-K approach, the classification accuracy was improved significantly. The proposed method is useful especially in case no malware families are available. For future work we propose investigating the idea of increasing N in N-gram selection, and applying dimensionality reduction methods on the input features to reduce the computational overhead added to the work as a result of increasing N.

References

1. Phelps, R.: Rethinking business continuity: emerging trends in the profession and the manager's role. J. Bus. Contin. Emerg. Plann. **8**(1), 49–58 (2014)
2. Mathur, K., Hiranwal, S.: A survey on techniques in detection and analyzing malware executables. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**(4), 422–428 (2013)

mmeybodi@aut.ac.ir

3. Idika, N., Mathur, A.P.: A Survey of Malware Detection Techniques. vol. 48, Purdue University (2007)
4. Bacci, A., et al.: Impact of code obfuscation on android malware detection based on static and dynamic analysis. In: 4th International Conference on Information Systems Security and Privacy. Scitepress (2018)
5. Vinod, P., Jaipur, R., Laxmi, V., Gaur, M.: Survey on malware detection methods. In: Proceedings of the 3rd Hackers' Workshop on Computer and Internet Security (IITKHACK 2009), pp. 74–79 (2009)
6. Urbanski, T.: Rapidshare & Co in the sights of the malware-mafia (2017)
7. Szor, P.: The Art of Computer Virus Research and Defense. Pearson Education (2005)
8. Cohen, F.: Computer viruses: theory and experiments. Comput. Secur. **6**(1), 22–35 (1987)
9. Annachhatre, C., Austin, T.H., Stamp, M.: Hidden Markov models for malware classification. J. Comput. Virol. Hacking Tech. **11**(2), 59–73 (2015)
10. Li, W.-J., et al.: Fileprints: identifying file types by n-gram analysis. In: Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop 2005, IAW 2005. IEEE (2005)
11. Weber, M., et al.: A toolkit for detecting and analyzing malicious software. In: Null. IEEE (2002)
12. Chinchani, R., Van Den Berg, E.: A fast static analysis approach to detect exploit code inside network flows. In: International Workshop on Recent Advances in Intrusion Detection. Springer (2005)
13. Rozinov, T., Rozinov, K., Memon, ND.: Efficient static analysis of executables for detecting malicious behaviors (2005)
14. Bilar, D.: Callgraph properties of executables. AI Commun. **20**(4), 231–243 (2007)
15. Ries, C.: Automated identification of malicious code variants (2005)
16. Bilar, D.: Opcodes as predictor for malware. Int. J. Electron. Secur. Digital Forensics **1**(2), 156–168 (2007)
17. Santos, I., et al.: Idea: opcode-sequence-based malware detection. In: International Symposium on Engineering Secure Software and Systems. Springer (2010)
18. Sung, A.H., et al.: Static analyzer of vicious executables (save). In: 20th Annual Computer Security Applications Conference 2004. IEEE (2004)
19. Shabtai, A., et al.: Detecting unknown malicious code by applying classification techniques on opcode patterns. Secur. Inf. **1**(1), 1 (2012)
20. Christodorescu, M., et al.: Malware Normalization. University of Wisconsin (2005)
21. Sgroi, M., Jacobson, D.: Dynamic and system agnostic malware detection via machine learning (2018)
22. Sathyanarayan, V.S., Kohli, P., Bruhadeshwar, B.: Signature generation and detection of malware families. In: Australasian Conference on Information Security and Privacy. Springer (2008)
23. Shankarpani, M., et al.: Computational intelligent techniques and similarity measures for malware classification. In: Computational Intelligence for Privacy and Security, pp. 215–236. Springer (2012)
24. Heaven, V.: Computer virus collection (2014). http://vxheaven.org/vl.php

mmeybodi@aut.ac.ir



RSS_RAID a Novel Replicated Storage Schema for RAID System

Saeid Pashazadeh(✉), Leila Namvari Tazehkand, and Reza Soltani

Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, East Azerbaijan, Iran
Pashazadeh@tabrizu.ac.ir

Abstract. Nowadays, due to the emergence of big data and its critical rule in most applications, data availability is big concern. For some applications, even the lack of a piece of information is not acceptable and have great drawbacks on results. Therefore, the storage reliability and guarantee of rapid data recovery is one of the main concerns. In case of disk failure, due to high storage volume, a lot of time is required for data recovery and this greatly decreases data availability. A new storage schema named RSS-RAID is presented in this paper. In this schema, disks are divided into groups with the same number of disks and data are stored as strips between disks with a particular algorithm based on a reversible hashing function. One advantage of proposed schema in comparison with similar models is that the location of the blocks is pre-known and when disk failure happens, number of missing blocks is clearly known and recovery algorithm do not need to search copies of missed blocks on the replica disks to recover them. This increases the recovery speed and causes more availability of data. Proposed schema is completely fault tolerant in case of one disk failure and fault tolerant against concurrent failure of up to three disks in the case that failed disks are located in the same group.

Keywords: RAID · Reversible hash function · Fault tolerancy · Fast recovery · Grouping

1 Introduction

Nowadays, most systems are computerized and they produce huge amount of accurate transactional data. These data are very valuable and gives more insights for future decisions. In many fields of science, education, medical treatment, trade and aerospace, systems are gathering large amount of valuable data and processing them. In many cases, loss of even a small portion of the data is not acceptable. Therefore, the storage and secure retrieval of data in the case of disk failures is one of the main challenges. By increasing the volume of data and their importance, storage and availability of the data is a challenging issue.

Today's storage systems are equipped with large capacity disks and due to the high volume of them, a lot of time is required to recovering the information in case of disk failure. If the number of defective disks increases, it will require plenty of time to recover the data from backup system and therefore decreases the availability of data. To

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 36–43, 2020.
https://doi.org/10.1007/978-3-030-37309-2_4

mmeybodi@aut.ac.ir

face with this problem, a new storage schema is presented in this paper. Disks are divided into groups with equal number of disks and data are stored as stripe between disks with a particular algorithm based on hashing. In addition to the original blocks, replication of blocks has been saved, and the main blocks and their replicas are stored based on predefined hash formula on disks. In case of disks failures, recovery algorithm knows the missing blocks' numbers and using hash function computes location of replicated copies and can find and recover the corrupted data blocks. The advantage of proposed models in comparison with similar models is that the location of data blocks is pre-known and will be retrieved without any search and this decreases data access and recovery time and increases data availability.

Is considered that the Redundant Array of Independent Disks (RAID) systems are used for data storing. After the disk failure, RAID retrieves the missing blocks to support data availability and reliability. In order to reduce the vulnerability and loss of data, the recovery process must be carried out quickly. Many researches have been done to improve the recovery speed. Xiang et al. [1, 2] proposed a hybrid recovery scheme to speed up the recovery process. Xu et al. [3] and Zhu et al. [4] used the similar approach to speed up single disk failure recovery for X-code and STAR code.

An architecture called RSS-RAID is introduced in this paper where the storage and recovery can be done with the help of reversible hashing-based formulas. There is no need to search corrupted data blocks for recovery purpose, and this lack of search reduces the recovery time.

2 Related Work

Data replication is mainly used approach to face with disk failure and its bad effects. Lee et al. [5] has proposed a double-layered architecture which uses erasure code to create redundancy in addition to the repetition of blocks. OI-RAID consists of two layers, outer layer and the inner layer. The outer layer is aligned with a disk grouping based on a complete graph, causing parallel I\O and increasing the recovery speed. Internal layer codes in each group of disks have increased reliability and have used both layers of RAID5 architecture for storage. This storage architecture provides quick recovery and high reliability.

Li et al. [6] has introduced a storage pattern called hybrid redundancy scheme plus computing (HRSPC). In this storage schema, both repetition and erasure code have been used to create redundancy. One of the advantages of HRSPC is that it uses little bandwidth to retrieve data and relatively reduces the cost of storage and increases reliability.

Zhu et al. [7] has proposed an alternative recovery algorithm that uses (greedy) hill climbing search technique to find a fast recovery solution. The main objective of their study is minimizing the overall time of recovery operation. Fundamental necessity for this purpose is reducing the amount of data read from live disks to recover. The (greedy) hill climbing search technique identifies the optimal solution for the recovery and replaces the current solution with optimal solution. This algorithm performs better than normal recovery for parallel-recovery architectures.

mmeybodi@aut.ac.ir

3 Proposed Method (RSS-RAID)

A novel storage algorithm for RAID systems is presented in this paper, which is based on the storing original data blocks and their replicas using hash function. This proposed storage architecture has following advantages, (1) the disks are grouped for parallel I\O execution like previous proposed architectures [5], (2) placement of data blocks and their replicas are done based on a hash function and this cause that when data access is needed, no search of data is required. The absence of a search operation greatly saves the time.

Summary of hash function for locating disk number and stripe of data block and its replica is presented in Table 1. n represents the number of available disks in the system and s indicates the total number of stripes. d_j denotes the j^{th} disk, b_i indicates the main data block i, m_i denotes replica of that block.

Table 1. Position of primary copy and replica of data block b_i.

Item	*Location*
disk number of primary data block i (b_i)	$i \bmod n$
disk number of replicated data block i (m_i)	$(((i \bmod n) + 1) \bmod n + ((i \operatorname{div} n) \bmod n)) \bmod n$
stripe number of primary data block i (b_i)	$(i \operatorname{div} n) + 1$
stripe number of replicated data block i (m_i)	$((i \operatorname{div} n) + (s \operatorname{div} 2)) + 1$

Figure 1 displays example of the RSS-RAID structure. Note that index numbers begin from zero and the number of disks n is 12 and the number of stripes s is 6. In this figure, placement of primary copy and replica blocks are based on the formulas of Table 1.

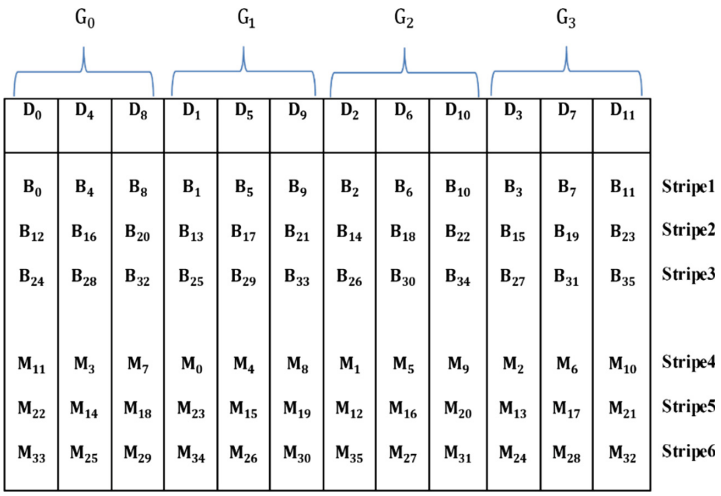


Fig. 1. An example of RSS-RAID for $n = 12$.

mmeybodi@aut.ac.ir

In case of a disc failure, all primary copies and replicas can be retrieved from other disks. Figure 2 shows how to retrieve the missing blocks during disk d_j failure. Left part of figure shows place of missed replica blocks and right hand side displays place of missed primary blocks. Place of each block is represented with a tuple with two fields; first field represents the disk number and second field represents stripe number.

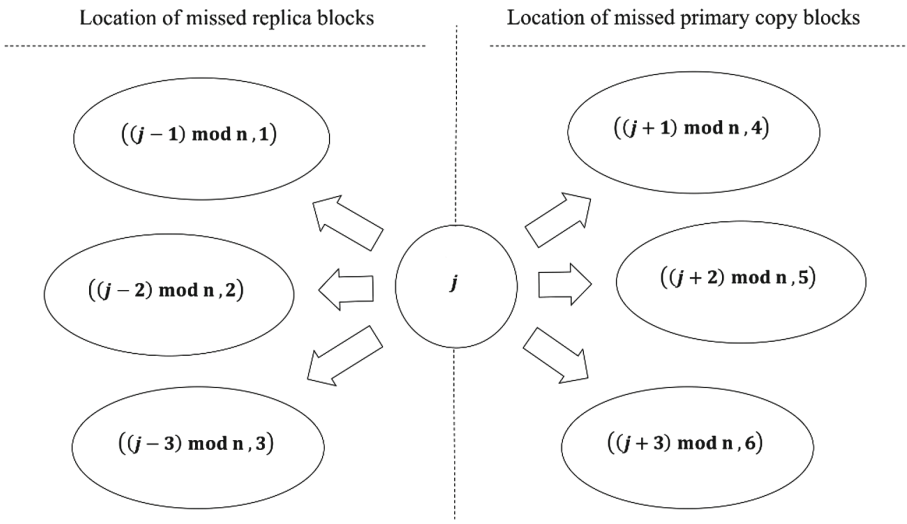


Fig. 2. Left side displays position where original version of missed replica blocks can be found and right side displays position where replica of missed data blocks can be found in case of disk d_j failure.

In Fig. 2 is not clarified that in case of disk j failure which primary copy blocks and which replica blocks in stripes will be missed. Table 2 shows that in case of disk j failure, which primary copy blocks and which replica blocks based on the stripe number will be missed. In this table index i denotes stripe number. Index number of primary copy blocks is less than or equal to s div 2 and index number of replica blocks is more than s div 2 and less than or equal to 2. Function ϑ is a recursive function as follows:

$$\vartheta(j,i) = \begin{cases} (j+n-1) mod\, n & i = 1 \\ (\vartheta(j,i-1)+n-1) mod\, n & i > 1 \end{cases}$$

Table 2. Index of missed primary copy and replica blocks in case of disk j failure. i denotes index of stripe.

Failure of disk index j	*Stripe number (i)*	*Index of missed block*
Primary copy missed blocks	$1 \le i \le (s\, div\, 2)$	$j + (i-1) * n$
Replicated missed blocks	$(s\, div\, 2) < i \le s$	$\vartheta(j, (i - s\, div\, 2)) + (i - (s\, div\, 2)) * n$

mmeybodi@aut.ac.ir

3.1 Relation Between Disks Number, Groups and Stripes

In the proposed model, disks are grouped with equal number of disks per group. The number of stripes is $2(x)$ where $0 < x \leq m$ and m is a natural number. For example in Fig. 1, for 12 disks and 4 groups, there can be 2, 4, 6 or any even number of stripes. In case of having data block more than $((s\,div\,2) * n)$, extension will perform based on the flowchart of Fig. 3 that will be discussed in Sect. 4.1.

As Fig. 1 displays, disks are grouped in $g = 4$ different groups. Interesting property of grouping disks is that in case of concurrent failure of 3 disks of one group, we can recover all data blocks using disks of one group. Disks are grouped based on the congruence class modulo mod g of disk's index.

3.2 Storage Algorithm

Pseudo code for storing data blocks and their replicas in the disks is as follows:

n= number of disks
L = list of blocks
s = number of stripes per disk
for i = 0 to the number of blocks in the list L do
PD(i) = L[i] mod n / PD(i) denotes disk number of primary copy of block i */*
PS(i) =(L[i] div n)+1 / PS(i) denotes stripe number of primary copy of block i */*
RD(i) = L[i] div n / RD(i) denotes disk number of replica of block i */*
RS(i) = ((L[i] div n)+s/2)+1/ RS(i) denotes stripe number of replica of block i */*
Store block i (b_i) at (PD(i), PS(i)) as primary and (RD(i),RS(i)) as replica
end for

3.3 Recovery Algorithm

In Sect. 3.2 detailed specification of proposed hash function is described. This hash function is one to one and therefore is reversible. In other words, reverse of this hash function is also a hash function. Reverse of hash function is used in case of disk failure to determine missed primary copy and replica blocks and also to determine that where we can find their copies on the other disks. Following pseudo code summarizes the recovery action to replace missed data blocks.

n = number of disks
j = index of failed disk
s = number of stripes per disk
for i = 1 to s div 2 do / i represents stripe number */*
missed primary copy block $j + (i-1) * n$ *is retrieved from stored replica located at* $((j+i) \bmod n\,,\, s\ div\ 2\ +\ i)$
end for
for i = (s div 2) +1 to s do / i represents stripe number */*
missed replicated block $\vartheta\big(j, (i - s\ div\ 2)\big) + \big(i - (s\ div\ 2)\big) * n$ *is retrieved from primary copy located at* $((j - (i - (s\ div\ 2)) \bmod n\,,\, i - (s\ div\ 2))$
end for

mmeybodi@aut.ac.ir

4 Analysis of RSS-RAID

In the previous proposed architectures presented at [5] and [8], usually standard RAID5 models for storing are used. The blocks are stored on a rotary basis and usually the location of the blocks will be random. However in the RSS-RAID model, since the storage of the main blocks and copies is based on hashing, the location of the blocks from the beginning is clear and does not require a search operation. Let assume disk 6 fails, as is shown in Fig. 1 since the storage algorithm is such that each primary copy has a replica and these two blocks not only never be stored in the same disks but also never be stored in the same group. So there is a copy of stored primary copy and replica blocks of disk 6 on the other live disks. Based on the proposed algorithm, there is no need for a search of lost blocks at recovery time, so the recovery time decreases.

In RSS-RAID model, because blocks are retrieved without searching, the data recovery time is reduced, thus increasing the recovery speed and increases the data availability.

4.1 Scalability and Fault Tolerance of RSS-RAID

Scalability of the system is key essential requirement. In the RSS-RAID model, if the number of disks and number of groups increases, we can store and retrieve blocks with the aforementioned algorithms. Flowchart of Fig. 3 displays method of storing data when number of blocks becomes more than $((s\ div\ 2) * n)$. Recovery of missed blocks will be performed based on the formulas of Table 2.

Ordered pair of storage in flowchart of Fig. 3 is like previous sections and first field represents the disk number and the second field represents the stripe number. Let assume that based on the flowchart of Fig. 3 we want to save block 36, the primary copy block will be stored in (0,4) and the replica block will be stored in (1, 8). As the number of blocks increases, the number of stripes will increase from 6 to 8.

Furthermore, high degree of system fault tolerancy causes more system reliability. RSS-RAID architecture is 100% fault tolerant against one disk failure and in this case, all stored blocks can be recovered from other disks. Let name this property as fault tolerancy of degree 1. If up to three failed disks are located in the same group, all data blocks are recoverable. But if the three concurrently collapsed disks are not at the same group, according to the number of collapsed disks, some blocks may lost.

mmeybodi@aut.ac.ir

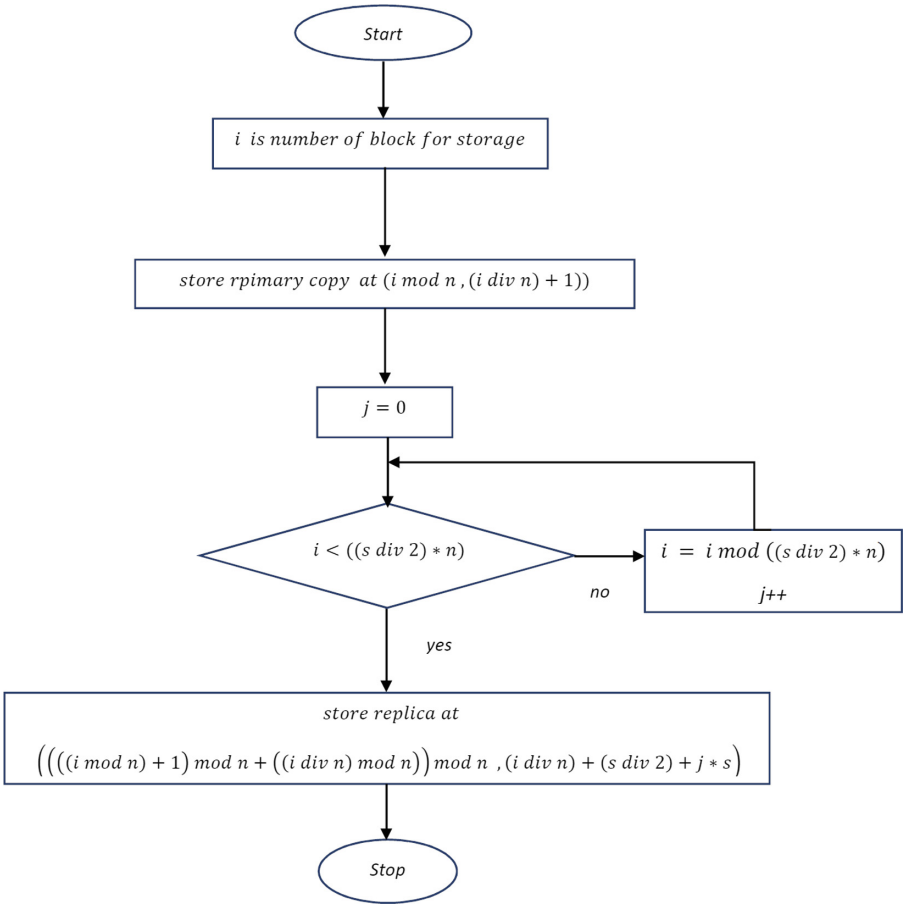


Fig. 3. Storage method in case of scalability.

5 Conclusion and Future Work

A storage model is presented using hashing technique named as RSS-RAID in this paper, which the original and copy blocks are stored as primary copy and replica. Each primary copy and its replica block are stored on a specified separate disks, such that never both of them may locate on a single disk. Also, this property is considered for groups. Therefore, never a primary copy and its replica will be stored on the same group of disks. In case of one disk failure all data can be recovered and so, proposed schema has fault tolerancy of degree one. It has fault tolerance of up to three in the case that failed disks are all located in the same group. Proposed hash function for storing data blocks is one to one and therefore its reverse also is hash function. This property causes that in case of recovery we do not need to search missed blocks in the other disks and place of missed data are known. So recovery speed of proposed schema is high in comparison with similar schemas.

mmeybodi@aut.ac.ir

For future works, in addition to redundancy, it is better to use erasure code storage to increase the fault tolerancy. Also, it is better to implement RSS-RAID in a real environment and compare the retrieval time with other storage models. The RSS-RAID model can also be modeled and evaluated by colored Petri net.

References

1. Xiang, L., Xu, Y., Lui, J., Chang, Q.: Optimal recovery of single disk failure in RDP code storage systems. In: Proceedings of ACM SIGMETRICS International Conference Measurement Modeling Computer Systems, pp. 119–130 (2010)
2. Xiang, L., Xu, Y., Lui, J., Chang, Q., Pan, Y., Li, R.: A hybrid approach to failed disk recovery using RAID-6 codes: algorithms and performance evaluation. ACM Trans. Storage **7**, 11 (2011)
3. Xu, S., et al.: Single disk failure recovery for X-code-based parallel storage systems. IEEE Trans. Comput. **63**(4), 995–1007 (2014)
4. Zhu, Y., Lee, P.P., Xu, Y., Hu, Y., Xiang, L.: On the speedup of recovery in large-scale erasure-coded storage systems. IEEE Trans. Parallel Distrib. Syst. **25**(7), 1830–1840 (2014)
5. Li, Y., Wang, N., Tian, C., Wu, S., Zhang, Y., Xu, Y.: A hierarchical RAID architecture towards fast recovery and high reliability. IEEE Trans. Parallel Distrib. Syst. **29**(4), 734–747 (2018)
6. Li, S., Cao, Q., Wan, S., Qian, L., Xie, C.: HRSPC: a hybrid redundancy scheme via exploring computational locality to support fast recovery and high reliability in distributed storage systems. J. Netw. Comput. Appl. http://dx.doi.org/10.1016/j.jnca.2015.12.012
7. Zhu, Y., Lee, P.P.C., Xu, Y., Hu, Y., Xiang, L.: On the speedup of recovery in large-scale erasure-coded storage systems. IEEE Trans. Parallel Distrib. Syst. **25**(7), 1830–1840 (2014)
8. Wan, J., Wang, J., Yang, Q., Xie, C.: S2-RAID: a new raid architecture for fast data recovery. In: Proceedings of IEEE 26th Symposium Mass Storage Systems Technologies, 3–7 May 2010

mmeybodi@aut.ac.ir



A New Distributed Ensemble Method with Applications to Machine Learning

Saeed Taghizadeh[1], Mahmood Shabankhah[2(✉)], Ali Moeini[2], and Ali Kamandi[2]

[1] Karlsruher Institut für Technologie, Karlsruhe, Germany
saeed.taghizadeh@kit.edu
[2] School of Engineering Science, College of Engineering, University of Tehran, Tehran, Iran
shabankhah@ut.ac.ir, {moeini,kamandi}@ut.ac.ir

Abstract. The main objective of this paper is to introduce a new ensemble learning model which takes advantage of the data which is originally distributed among a group of local centers. In this model, we first train a group of client nodes which have access only to their own local data sets. High classification rate is not required in this phase. In the second phase, the master node learns which client nodes are more likely to classify correctly a given data instance. Therefore, only the responses of these effective nodes will be used in the classification step. A major advantage of our algorithm, as the experimental results confirm, is that the network can obtain high classification rates by using a relatively small fraction of the whole data set. Moreover, this learning scheme is fairly general and can be employed in other contexts as well.

Keywords: Ensemble methods · Machine learning · Distributed learning · AdaBoost · Big data

1 Introduction

Rapid growth in the amount of the data generated worldwide every single moment, has brought new challenges to the application of machine learning algorithms designed to extract useful information from this huge basin. Standard machine learning algorithms do not, in general, perform well in such situations. One of the most effective ways to tackle the problem of data volume is to use *ensemble methods* [9,20]. The key idea is to assign the whole data set or smaller fractions of it to different learning systems. The results of these systems are then combined in some way or another to build a model which can handle effectively and efficiently very huge data sets. Algorithms like Bagging [1], ADABOOST [8], random forests [2] are just a few examples of such learning methods.

Another motivation for using ensemble methods is that in most applications the data is itself distributed among different data centers. As a consequence, to

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 44–58, 2020.
https://doi.org/10.1007/978-3-030-37309-2_5

mmeybodi@aut.ac.ir

process the entire data set by a single machine is either impossible or computationally very costly. This situation occurs often in modern real world applications, and brings with itself a new paradigm called *distributed learning*. Ensemble methods also can be cast into the framework of distributed machine learning. A good distributed machine learning algorithm should meet the following criteria:

- Has high accuracy;
- Has low execution time;
- Supports incremental learning
- Supports dynamic learning

To reach these objectives, we devised a model in which the task of learning is carried out in several phases. The main idea has been to exploit the results of each phase in order to improve learning in subsequent stages. The model we propose consists of a master node along with several client nodes. In the first phase of learning, client nodes are trained based on the local data they have access to. This phase can be run in parallel which reduces the total execution time. Once the training of client nodes has been completed, the master node is trained. The purpose of this second phase for the master node is to learn, based on the results of the first phase, which client nodes are more likely to produce the correct response to a given input data. Combined together, these two phases improve greatly the performance of the system. Experimental results on Optical Digits database confirm our claim.

The paper is organized as follows. In Sect. 2 we introduce briefly ensemble and distributed learning methods. We then proceed to introduce our model and its learning algorithm in Sect. 3. The results of experiments on Optical Digits database Sect. 4. Conclusions and some future directions are discussed in Sect. 5.

2 Background of Ensemble Methods

Suppose we have a set of points $\mathbf{x}_i \in \mathcal{R}^n (i = 1, \ldots, N)$ belonging to one of the K classes $C_j \, (j = 1, \ldots, K)$. We use 1-of-$K$ encoding scheme to represent the target vectors. More precisely, if $\mathbf{x} \in C_j$, then the corresponding target vector $\mathbf{y}$ is the K-dimensional vector whose j-th coordinate is 1 and all other elements are zero. We now consider the data set $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$. This data set is used to train L different classifiers, denoted by $h_1, \ldots, h_L$, each of which having its own learning algorithm. Once the learning phase is completed, a test point $\mathbf{x}$ will be given as input to these classifiers. Since the outputs are not necessarily the same, the target vector $\mathbf{y}$ can be predicted by combining, in a certain way, the predictions of all classifiers $h_1, \ldots, h_L$. This is technically known as an *ensemble method*.

By definition, ensembles are sets of classifiers wherein the predictions of all classifiers are somehow combined (e.g. majority voting) to classify a test sample. This usually improves the overall performance of the ensemble compared to individual classifiers. There are three major ensemble methods, namely *Bagging*, *AdaBoost* and *random forests*. In the following we give a brief description of how each method works.

mmeybodi@aut.ac.ir

2.1 Bagging

Bagging is a simple and popular ensemble method which was proposed by Breiman [1]. It helps improve the accuracy and stability of learning algorithms. Roughly speaking, bagging can be viewed as a *model averaging* method.

Suppose we have a training set $\mathcal{D}$ of size d. We create k data sets $\mathcal{D}_i$, $(i = 1, \cdots, k)$, by uniformly sampling (with replacement) from $\mathcal{D}$. Each $\mathcal{D}_i$ is called a *bootstrap sample*. $\mathcal{D}_i$ is then used to train a model M_i $(i = 1, \cdots, k)$. The output of the composite model M_* to a new sample $\mathbf{x}$ is obtained by taking the majority vote among the predicted class of $\mathbf{x}$ using all models M_i, $(i = 1, \cdots, k)$. The algorithm is summarized in Fig. 1.

Algorithm: Bagging. The bagging algorithm—create an ensemble of classification models for a learning scheme where each model gives an equally weighted prediction.

Input:

- D, a set of d training tuples;
- k, the number of models in the ensemble;
- a classification learning scheme (decision tree algorithm, naïve Bayesian, etc.).

Output: The ensemble—a composite model, $M*$.

Method:

(1) **for** $i = 1$ to k **do** // create k models:
(2) create bootstrap sample, D_i, by sampling D with replacement;
(3) use D_i and the learning scheme to derive a model, M_i;
(4) **endfor**

To use the ensemble to classify a tuple, X:

let each of the k models classify X and return the majority vote;

Fig. 1. Bagging algorithm [9].

2.2 AdaBoosting

AdaBoost was proposed by Freund and Schapire [8]. It has mainly been used in classification problems where a set of weak classifiers are combined to form a stronger one. In particular, this method has been successfully applied in decision tree induction (Quinlan [14]) and naïve Bayesian classification (Elkan [5]).

Suppose that $\mathcal{D}$ is a training set of size d. The first data set $\mathcal{D}_1$ is created by uniformly sampling (with replacement) from $\mathcal{D}$. Since the sampling is uniform, we can actually imagine that all samples are assigned an equal weight (or probability) $1/d$. The set $\mathcal{D}_1$ can now be used to train the first model M_1. The key

mmeybodi@aut.ac.ir

idea in AdaBoost is to create *iteratively* a series of classifiers, M_i $(i = 1, \cdots, k)$. After M_i is trained, we assign new weights (probabilities) to training samples in such a way that misclassified samples are assigned higher weights whereas the weights of the correctly classified samples is decreased. As a result, misclassified samples will appear in the subsequent training set $\mathcal{D}_{i+1}$ with higher probability. On the hand, after each M_i is trained, AdaBoost assigns a weight to it as well. This weight is a function of M_i's accuracy. More precisely, the weight w_i is given by

$$w_i = \log \frac{1 - \text{error}(M_i)}{\text{error}(M_i)}.$$

Therefore, more accurate models are assigned higher weights. This makes sure that better models will have a stronger impact on the final outcome of the composite model M_*. Indeed, to classify a new sample $\mathbf{x}$, we sum the weights of those classifiers that assigned $\mathbf{x}$ to a given class c. The class having the highest sum will be considered as the predicted class of $\mathbf{x}$. See Fig. 2 for the details of AdaBoost algorithm [8].

AdaBoost algorithm has been the subject of numerous theoretical and practical studies. We only mention a few here. In [17], the authors introduce a particular form of sampling called *weighted novelty selection*, which combined with standard AdaBoost, leads to significant speed up of the learning process at the expense of very little reduction in the overall accuracy. In another work [21], AdaBoost has been combined with information from YCbCr color space to find a new face detection algorithm in still images. As another application, the authors in [19] propose a sort of cascaded SVM architecture based on AdaBoost boosting. Their results show an improvement in the classification accuracy of the classical SVM algorithm. A fuller discussion of theoretical models to analyze AdaBoost and extensions to multiclass problems along with some future research work are studied in [3].

2.3 Random Forest

The last ensemble method considered in this section is random forests which is described by Breiman [2]. Symbolically, if the set of classifiers in our ensemble consists only of decision trees then the collection may be viewed as a forest. Bagging method is used to train the decision trees in a random forest.

Given a training set $\mathcal{D}$ of size d, multiple data sets $\mathcal{D}_i$, $(i = 1, \cdots, k)$, are created by uniformly sampling (with replacement) from $\mathcal{D}$. Each $\mathcal{D}_i$ is then used to train a decision tree M_i $(i = 1, \cdots, k)$. Random forest adds some randomness to the training of each M_i. Instead of finding the best splitting feature among all features, M_i uses a fraction f of features at each node to grow the tree. Once all trees M_i $(i = 1, \cdots, k)$ are constructed, the composite model M_* combines the responses of M_i to a new sample $\mathbf{x}$ to find the class with the highest probability.

mmeybodi@aut.ac.ir

Algorithm: AdaBoost. A boosting algorithm—create an ensemble of classifiers. Each one gives a weighted vote.

Input:

- D, a set of d class-labeled training tuples;
- k, the number of rounds (one classifier is generated per round);
- a classification learning scheme.

Output: A composite model.

Method:

(1) initialize the weight of each tuple in D to $1/d$;
(2) **for** $i = 1$ to k **do** // for each round:
(3) sample D with replacement according to the tuple weights to obtain D_i;
(4) use training set D_i to derive a model, M_i;
(5) compute $error(M_i)$, the error rate of M_i (Eq. 8.34)
(6) **if** $error(M_i) > 0.5$ **then**
(7) go back to step 3 and try again;
(8) **endif**
(9) **for** each tuple in D_i that was correctly classified **do**
(10) multiply the weight of the tuple by $error(M_i)/(1 - error(M_i))$; // update weights
(11) normalize the weight of each tuple;
(12) **endfor**

To use the ensemble to classify tuple, X:

(1) initialize weight of each class to 0;
(2) **for** $i = 1$ to k **do** // for each classifier:
(3) $w_i = log \frac{1-error(M_i)}{error(M_i)}$; // weight of the classifier's vote
(4) $c = M_i(\boldsymbol{X})$; // get class prediction for $\boldsymbol{X}$ from M_i
(5) add w_i to weight for class c
(6) **endfor**
(7) return the class with the largest weight;

Fig. 2. AdaBoost algorithm [9].

3 Proposed Model Based on Distributed Systems

In this section we introduce our own basic model: DYnamic Adaptive Boosting (DYABoost algorithm). We consider a group of systems (nodes) where each node takes part in the learning process. in the learning process. The pattern of connections among these nodes is as in Fig. 3. As we can see, there is a master node along with some other client nodes. Client nodes can exchange information with the master node and vice versa. However, there is no connection between client nodes. This reduces the bandwidth needed during the execution stage. In addition, client nodes have no access to others' local data.

mmeybodi@aut.ac.ir

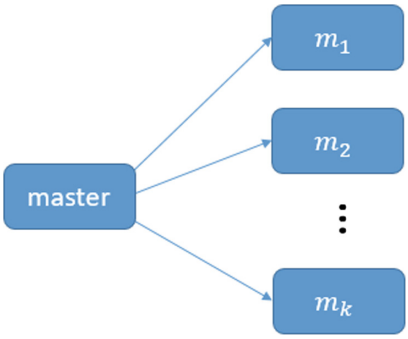


Fig. 3. Distributed dynamic AdaBoost component model

The basic training algorithm consists of two phases, Phase I and II. In the first phase, only the client nodes are trained. Because of the system's architecture, this phase can be done in parallel which significantly reduces the total learning time. In the second phase, master node starts learning via a specific interaction with client nodes.

After introducing this basic model, we make some modifications in order to turn it into an incremental learner. In this part, we use Learn++ algorithm as the core of learning in our model. This leads to a new method that we call "DYABoost algorithm". In this case, we are able to avoid sequential operations that take place in Learn++ algorithm. Indeed, distributed Learn++ uses the ideas and patterns of distributed systems for parallel learning and therefore has better performance in comparison with Learn++ algorithm. In addition, this approach enables us to use feedbacks which allows incremental learning. Another feature of this algorithm, as the experiments show, is that it can reach high accuracy by using fewer training examples. To the best of our knowledge, this learning scheme has not been previously introduced in the literature.

3.1 Phase I

In the first phase of our algorithm, each client node m_ℓ ($\ell = 1, \ldots, K$) in the system is trained based on its own learning algorithm and using its local data (See Fig. 4(a)). Note that the master node is not trained in this phase. Only its local data is being assigned.

Since the training algorithm of each client node is up to itself, any classification algorithm (e.g. support vector machines [4,10,16,18], decision trees [12,13], naïve Bayes [9], KNN [7], neural networks [6,15], etc.) may be used at this stage. In addition, this phase can be done in parallel among all nodes because of the system's architecture. It should be emphasized that the learning algorithms used in this phase are all weak learning algorithm. A *weak classifier* is a classifier whose misclassification rate is no more than 50%. For a two class problem, however,

mmeybodi@aut.ac.ir

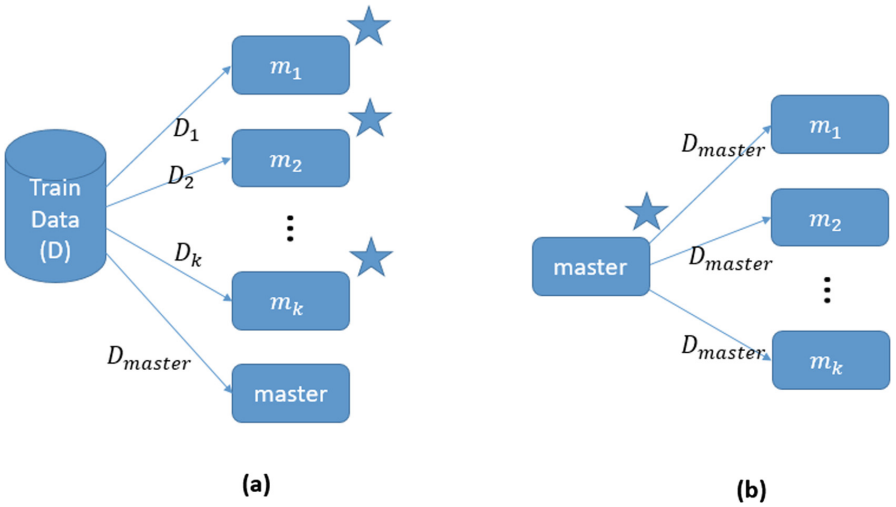


Fig. 4. Distributed dynamic AdaBoost learning process

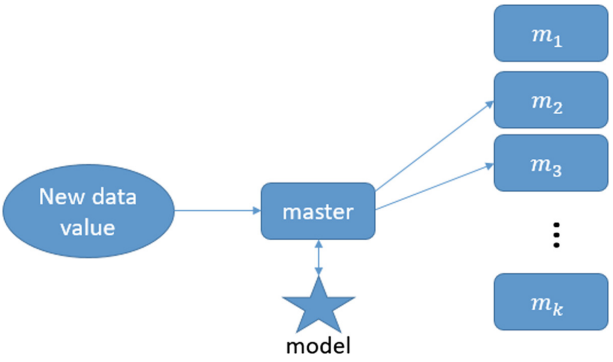


Fig. 5. Distributed dynamic AdaBoost test Process

this is indeed the minimum achievable if the data are simply assigned into the classes randomly. One motivation for using weak classifiers in our model is to avoid over-fitting issues.

mmeybodi@aut.ac.ir

Algorithm 1. Dynamic Adaptive Boosting **(DYABoost)** Algorithm

Input: **D**, a set of d class-labeled training tuples
K, the number of classifiers
Output: a composite model.
procedure DYABoost
    partition D into $K+1 parts : D_1, \ldots, D_K, D_{master}$
    foreach classifier m_ℓ $(\ell = 1, \ldots, K)$ **do**
        $Train(m_\ell, D_\ell)$;
    end foreach
    $D_{master} \leftarrow \{(\mathbf{x_i}, \mathbf{c_i}) : \mathbf{x_i} \in \mathcal{R}^n, \mathbf{c_i} \in \mathcal{R}^{\mathcal{M}}, (i = 1, \ldots, N)\}$
    foreach $\mathbf{x} \in D_m$ **do**
        foreach machine(classifier) m_ℓ $(\ell = 1, \ldots, K)$ **do**
            $t \leftarrow Test(m_\ell, \mathbf{x})$;
            if $t = class(\mathbf{x})$ **then**
                $\delta_\ell(\mathbf{x}) \leftarrow 1$;
            else
                $\delta_\ell(\mathbf{x}) \leftarrow -1$;
            $\mathbf{t}(\mathbf{x}) \leftarrow (\delta_1(\mathbf{x}), \ldots, \delta_K(\mathbf{x}))$
        end foreach
    end foreach
    $D_M \leftarrow \{(\mathbf{x}_i, \mathbf{t}(\mathbf{x}_i)), i = 1, \ldots, N\}$
    $Train(Master, D_M)$.

Algorithm 2. Dynamic Adaptive Boosting **(DYABoost)** Algorithm

Input: **x**, test tuple
K, the number of classifiers
Output: predicted class.
procedure DYABoostTest
    $\mathbf{y} \leftarrow Test(Master, \mathbf{x})$
    for $i : 1 \ldots K$ **do**
        if $y_i = 1$ **then**
            $t \leftarrow Test(V_i, \mathbf{x})$;
    $c \leftarrow majority(t)$;
    return c;

3.2 Phase II

The second phase of our algorithm involves training the master node (See Fig. 4(b)). Let $\mathbf{x}_i \in \mathcal{R}^n (i = 1, \ldots, N)$ be the training data for the master node. Each input $\mathbf{x}_i$ is already labeled and belongs to one of the classes C_j $(j = 1, \ldots, \mathcal{M})$. We then proceed as follows.

1. The input vector $\mathbf{x}$ is given to the master node.
2. Master node sends $\mathbf{x}$ to all of the client nodes which were trained in Phase I.
3. The classification results of the client nodes are then sent back to the master node. We then form a new vector, denoted by $\mathbf{t}(\mathbf{x}) \in \mathcal{R}^]mathcalM$, which

mmeybodi@aut.ac.ir

encapsulate the responses of the client nodes. More precisely, we put

$$\mathbf{t}(\mathbf{x}) = (\delta_1(\mathbf{x}), \ldots, \delta_m(\mathbf{x})),$$

where

$$\delta_\ell(\mathbf{x}) = \begin{cases} 1, & \text{if } m_\ell \text{ classifies } \mathbf{x} \text{ correctly;} \\ -1, & \text{otherwise.} \end{cases}$$

4. The vector $\mathbf{t}(\mathbf{x})$ is then considered as the target vector for $\mathbf{x}$. In other words, the set

$$\mathcal{D}_M = \left\{ \left(\mathbf{x}_i, \mathbf{t}(\mathbf{x}_i)\right), i = 1, \ldots, N \right\},$$

will be used as the training data for the master node (Algorithm 1). Intuitively, the purpose of this step is to learn, for a given input $\mathbf{x}$, which nodes are more likely to produce the correct classification.

The details of Phase II are illustrated in Algorithm 1.

3.3 Testing Stage

Once Phase I and II are completed as described above, the system is ready to be tested on new instances (see Fig. 5). The application procedure is as in Algorithm 2. Given a test input $\mathbf{x}$, the master node generates its output vector $\mathbf{t}(\mathbf{x})$. We then select those components of $\mathbf{t}(\mathbf{x})$ whose value is 1. The corresponding client nodes are more likely to classify $\mathbf{x}$ correctly. Therefore, the master node sends $\mathbf{x}$ as a test data to these client nodes only, and receives their predictions concerning the true class of $\mathbf{x}$. At last, a voting scheme in the master node will determine the predicted class of $\mathbf{x}$. In case no component of $\mathbf{t}(\mathbf{x})$ is 1, the voting scheme should be carried out among all client nodes m_ℓ ($\ell = 1, \ldots, m$).

3.4 Modifications

To implement our learning algorithm, we first had recourse to simple multilayer perceptron networks. However, the results obtained were not so satisfying. Even in this case we observed that our algorithm had a better performance compared to the case where only a single machine was used. In order to fully exploit the distributed nature of the data, we used some of the ideas of the *Learn* + + algorithm [11]. Learn++ algorithm has some features which make it a good choice to be used as the learning core of our algorithm. For instance, Learn++ is able to detect new unforseen classes among the training data. In addition, because of its incremental learning nature, as the training process continues with newly arrived data, the system does not forget the data already learned. In applications based on Learn++ which we consider here, classification is carried out via a weighted majority voting scheme among client nodes trained in Phase I. Moreover, since Learn++ adapts well to incremental learning environments, one could even consider the case where new client nodes are added in the middle of the training of the master node. This last case, however, is not considered here and may be the subject of another study.

mmeybodi@aut.ac.ir



Fig. 6. Sample characters from OpticalDigits database

4 Experimental Results

To test the performance of our learning algorithm, we considered the problem of handwritten character recognition. As for the input data, we used OpticalDigits database available in the machine learning repository of UCI[1]. Figure 6 shows a few examples of the characters in this database. In fact this data set comprises a total of 5620 handwritten samples of digits 0 to 9 stored in the form of 8×8 matrices. Out of this set, 1400 samples were randomly chosen as training data. We then used 1000 data points, divided into five groups of equal size, to train five client nodes, and 400 samples to train the master node. The remaining 4220 data samples were used as test set.

To train the master and client nodes, we first used basic multilayer perceptron network as the core of our model. Since the results were not so promising, we turned to Learn++ as the core algorithm in the training of our MLP networks.

[1] https://archive.ics.uci.edu/ml/datasets/optical+recognition+of+handwritten+digits.

mmeybodi@aut.ac.ir

We then saw a great improvement in the results, and the combination of our model with ideas from Learn++ proved to be very successful. We first explain the basic MLP network approach.

The MLP network architecture we used in each client node had 64 input units, 30 units in the hidden layer, and 10 units for output layer to represent the class of the input data. The activation function for hidden and output units was tanh function. We initialized the weights to small random numbers, and continued the training until the total error was less than 0.3. This error rate was chosen, by cross-validation, to make our client nodes into weak learners. Figures 7, 8, 9, 10 and 11 show the weight histogram and total network error for each client node.

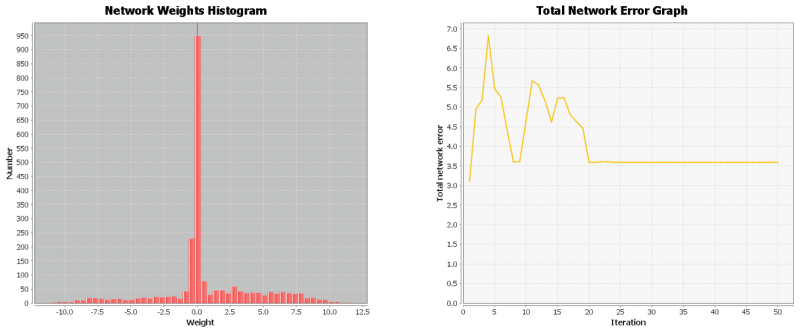


Fig. 7. Machine1

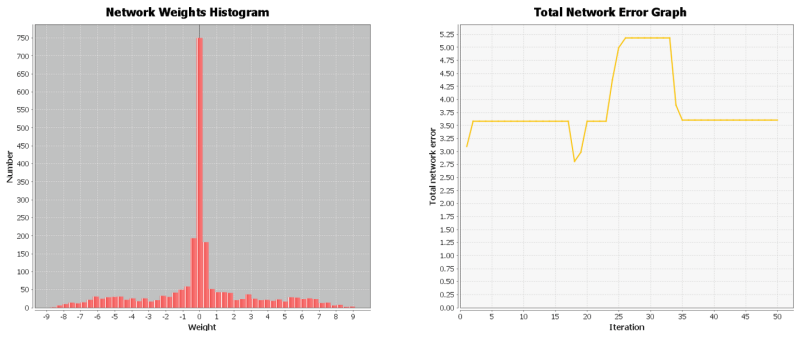


Fig. 8. Machine2

This constituted Phase I of our algorithm. It can be seen that, the weights of each network had not in fact changed much during the training, and remained largely near zero. This was due to the fact that we had set a relatively high error rate as the stopping condition for the training of MLP nets.

mmeybodi@aut.ac.ir

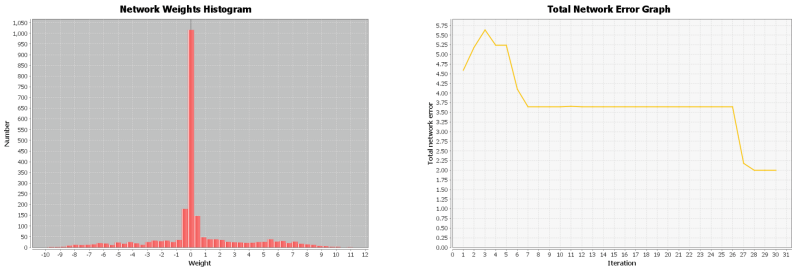


Fig. 9. Machine3

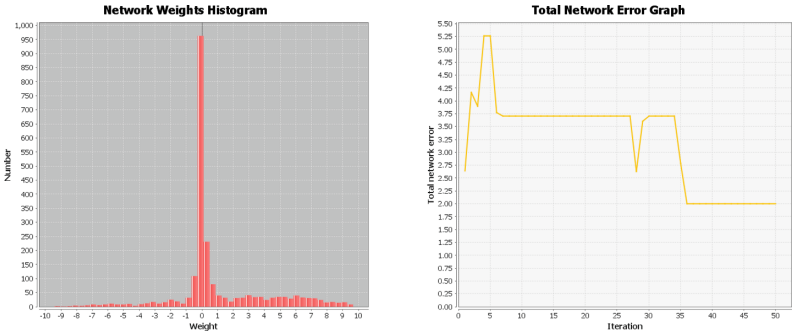


Fig. 10. Machine4

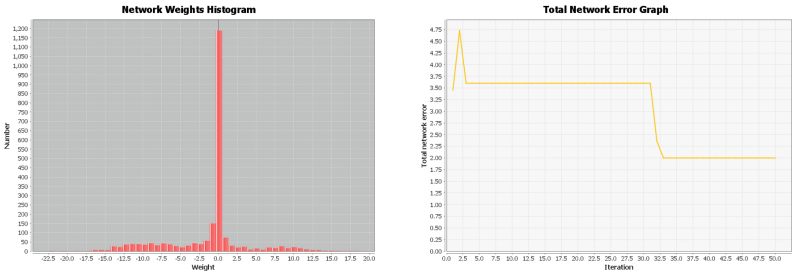


Fig. 11. Machine5

We then started training the master node which constituted the second phase of our algorithm. The MLP network architecture for the master node had 64 input units, 30 units in the hidden layer, and 5 output units. Note that the training data for the master node comprised 400 instances of the form $\big(\mathbf{x}, \mathbf{t}(\mathbf{x})\big)$ where $\mathbf{t}(\mathbf{x})$ was a five-dimensional vector formed from the responses of the client nodes to the input $\mathbf{x}$ (see Phase II). Weight histogram and total training error are shown in Fig. 12.

mmeybodi@aut.ac.ir

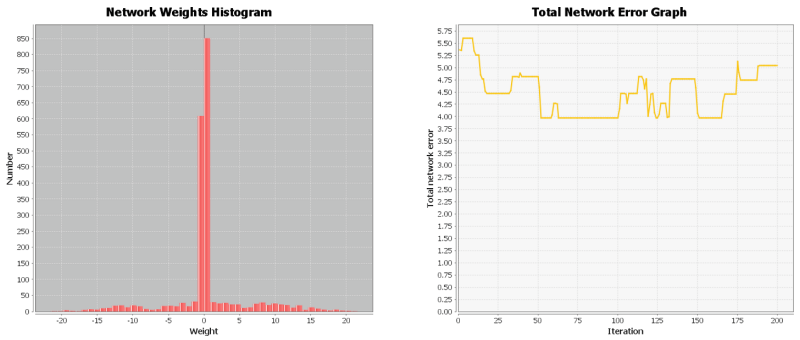


Fig. 12. Master machine

This distributed model was tested with 4220 test samples. However, the classification rates we obtained were not so encouraging. It showed that using basic classifiers like MLP in our model is not sufficient to get results.

To overcome this problem we turned to Learn++ as the core of learning model keeping the MLP nets with the same architecture as before. We implemented Learn++ for each node setting $K = 1$ and $T_1 = 30$. In other words after training with Learn++ we will have 30 weak hypothesis. Table 1 shows the classification rates for the client nodes when tested on the 400 training instances of the master node.

Table 1. Performance of client nodes tested on the training data for the master node

Machine name	m_1	m_2	m_3	m_4	m_5
Number of correct classification	361	354	348	352	352
Number of validation instances	400	400	400	400	400
Accuracy	0.9025	0.885	0.87	0.88	0.88

As we see, the average classification rate of client nodes is about 88% which is much higher than the basic MLP model. One might guess that this would lead to over-fitting problems in the test stage. That this is not the case can be verified from Table 2 which shows the performance of the whole system on the test set.

Table 2. Comparison of the proposed model results with the single machine model

Machine name	Proposed model	Single machine
Number of correct classification	3768	2720
Number of all instances	4220	4220
Accuracy	89.2	64.45

mmeybodi@aut.ac.ir

5 Conclusion and Future Works

We introduced a distributed learning algorithm which consisted of a master node along with several client nodes. Client nodes are directly connected to the master node, and each has its own local data. The ultimate goal is to somehow bring together the information which is distributed in these local data centers. To this end, we devised a new distributed learning algorithm which runs in two phases. In the first phase only the client nodes are trained, whereas in the second phase, the master node is trained via a special interaction with client nodes.

To put our algorithm into the test, we considered the problem of handwritten character recognition. As for the training of the system, we first considered basic MLP networks. In a further step and to improve the performance, we incorporated ideas from Learn++ algorithm into our own algorithm. We then observed that this way we can get classification rates of up to 90% by using only a relatively small fraction of the data as training data. It proves that, our model is capable of exploiting the knowledge of each node without allowing direct data transmission between client nodes. This, in our opinion, is a great advantage of our algorithm.

There remain other extensions which can be studied in the future. Some of them are provided below:

- Using probabilistic methods in the training of client and master nodes.
- Modifying the structure of our model in order to support algorithms such as SVM in distributed environments.
- Replacing Learn++ with other incremental algorithms (e.g. ADABOOST) in the core of our model.
- Doing experiments in other databases to test the performance of our model in various classification problems.

References

1. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
2. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
3. Cao, Y., Miao, Q.G., Liu, J.C., Gao, L.: Advance and prospects of adaboost algorithm. Acta Automatica Sinica **39**(6), 745–758 (2013). https://doi.org/10.1016/S1874-1029(13)60052-X
4. Cristianini, N., Shawe-Taylor, J., et al.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
5. Elkan, C.: Boosting and Naive Bayesian learning. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (1997)
6. Fausett, L.V., et al.: Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, vol. 3. Prentice-Hall, Englewood Cliffs (1994)
7. Fix, E., Hodges, J.L.: Discriminatory analysis, nonparametric discrimination: consistency properties. Technical report 4, USAF School of Aviation Medicine, Randolph Field, Texas (1951)

mmeybodi@aut.ac.ir

8. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
9. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann (2011)
10. Herbrich, R.: Learning Kernel Classifiers: Theory and Algorithms (adaptive Computation and Machine Learning). MIT Press, Cambridge (2002)
11. Polikar, R., Upda, L., Upda, S.S., Honavar, V.: Learn++: an incremental learning algorithm for supervised neural networks. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **31**(4), 497–508 (2001)
12. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
13. Quinlan, J.R.: C4. 5: Programming for Machine Learning, vol. 38, p. 48. Morgan Kauffmann (1993)
14. Quinlan, J.R., et al.: Bagging, boosting, and C4.5. In: AAAI/IAAI, vol. 1, pp. 725–730 (1996)
15. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (2007)
16. Schölkopf, B., Smola, A.J., et al.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
17. Seyedhosseini, M., Paiva, A., Tasdizen, T.: Fast adaboost training using weighted novelty selection. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1245–1250 (2011). https://doi.org/10.1109/IJCNN.2011.6033366
18. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (2013)
19. Wang, R.: Adaboost for feature selection, classification and its relation with SVM, a review. Phys. Procedia **25**, 800 – 807 (2012). https://doi.org/10.1016/j.phpro.2012.03.160, International Conference on Solid State Devices and Materials Science, 1–2 April 2012, Macao
20. Webb, A.R.: Statistical Pattern Recognition. Wiley, Hoboken (2003)
21. Xu, J., Goto, S.: Proposed optimization for adaboost-based face detection. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 8009 (2011). https://doi.org/10.1117/12.896293

mmeybodi@aut.ac.ir

A Glance on Performance of Fitness Functions Toward Evolutionary Algorithms in Mutation Testing

Reza Ebrahimi Atani(✉), Hasan Farzaneh, and Sina Bakhshayeshi

Department of Computer Engineering, University of Guilan,
P.O. Box, 3756, Rasht, Iran
rebrahimi@guilan.ac.ir, hassan.9361@gmail.com,
sina.bakhshayeshi@gmail.com

Abstract. Nowadays, Internet and web applications have influenced on different aspects of human life. Therefore there are always some needs to different software platforms for implementation of electronic commerce or electronic governance. Hence a great market is now devoted to software production in various platforms. Regarding such market demand, producing high-quality softwares with reliability, safety and availability services are considered as an important issue. To be more specific all software companies use software testing concepts as an independent process in software development cycle. There are various methods for software testing, but mutation testing is one of the most powerful tools. In mutation testing, high-quality test-case generation plays a key role and it has a direct relation with quality of software testing. There are different techniques for test-case generation where evolutionary algorithms are among the most common ones. Since each evolutionary algorithm needs an appropriate fitness function which is dependent on target problem, it is very important to know that for each evolutionary algorithm which fitness function generates better test cases. The main goal of this paper is to answer this question and a treatment of five evolutionary algorithms regarding four different fitness functions are classified in this work.

Keywords: Mutation testing · Test-case generation · Fitness function · Evolutionary algorithm

1 Introduction

Regarding the revolutionary expansion occurred by information technology in our world, many changes have occurred in people's daily life. Web applications, mobile social networks and commercial and industrial softwares are as part of these rapid change factors in human life. According to the key role of softwares in human life, production of high-quality products is considered as an important goal of private sector. To be able to achieve this goal, all software companies hire software testers and try to apply software testing concepts and tools. In software testing, high-quality test-case generation plays a key role because it has a direct relation with quality of test. In other words, whatever the used test-cases have higher-quality, the software testing will have

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 59–75, 2020.
https://doi.org/10.1007/978-3-030-37309-2_6
mmeybodi@aut.ac.ir

a higher potential and it will also detect more faults. As a result, most testers have a strong focus on topic of test-case generation. Software testing is a very broad concept but Mutation Testing is one the most powerful tools among them. Mutation Testing is a fault-based method and was first introduced in 1971 in a student paper by Lipton [1]. Subsequently, it was officially introduced in 1978 by DeMillo [2] and Hamlet [3].

Generally, mutation testing first generates different copies of main program. Then, it injects various faults within them using mutation operators. Mutation operators play a key role in mutation testing. Whatever mutation operators of mutation testing are precisely designed, output test-cases of mutation testing will have higher potential and more likely will detect more faults of software. These faulty software copies are called *Mutant*. After generating mutants, mutation testing using various techniques attempts to generate test-cases that are able to detect the injected faults. To be able to detect the faults, results of main program and mutants are compared together. If there is any difference in two results, the mutant is considered as *killed* mutant but otherwise it is considered as *live* mutant.

In recent years, extensive researches have been done on mutation testing. According to the researches, many scientists have concluded that mutation testing is more powerful than other techniques [4]. In addition, Frankel et al. [5] and Offutt et al. [6] also proved that mutation testing is a much more successful than other techniques in detecting faults of software.

Mutation testing has several research topics, but one the most important of the topics is high-quality test-cases generation. A test-case is considered as a high-quality test-case in mutation testing when it is able to detect all or maximum number of the injected faults of mutants. High-quality test-cases not only reduce computation cost of mutation testing, but they are also the best option to test a software because they have a high potential and more likely will detect faults of a software under test. Anyway, one useful technique for high-quality test-case generation is Evolutionary Testing (ET) algorithms. ET attempts to generate high-quality test-case using diverse Evolutionary Algorithms (EA) such as Genetic Algorithm (GA), Hill climbing (HC) and etc. As we all know and is explained in literature, structure of EAs is such that it depends on fitness functions. In fact, fitness functions are responsible for guiding EAs in the search space. Considering key role of fitness functions, selection of an appropriate fitness function for EAs in mutation testing is very important because it does not only leads to quick guidance in the search space, but also plays a key role in high-quality test-case generation. However, a question that arises here is that which EA with which fitness function does generate higher-quality test-cases? To be able to partially answer the question, the paper aims to examine treatment of five EAs with four fitness functions. The main contribution and innovations of the paper is as follows:

- Use of Queen (QA) and Particle Swarm Optimization (PSO)
- Introducing RDIFF fitness function
- Comparing treatment of EAs (PSO, Genetic (GA), Queen (QA), Bacteriological (BA), Hill climbing (HC)) with different fitness functions (MS, APP, RDIFF, BR) in both weak and strong mutations.

The rest of the paper is organized as follows: Sect. 2 presents literature review in recent advances in mutation testing. Section 3 provides basic definitions and

mmeybodi@aut.ac.ir

background information about mutation testing. Section 4 describes the applied evolutionary algorithms and fitness functions. Section 5 presents the simulation set up and experimental results. Finally, in Sect. 6 further discussions and future work are described and the paper concluded.

2 Related Works

In this section of the paper it is tried to survey the most recent works presented regarding mutation testing, test case generation.

2.1 Mutation Testing

Initial generation of mutation testing tools was interpreted-based technique. In the technique, output of a mutant was directly interpreted from source code. Offutt and King [14] improved the technique. In their research, a program is translated into an intermediate code level in FORTRAN. Main cost of the technique was defined base on interpretation cost of source code. The technique is convenient for small programs and has high flexibility. Subsequently, they designed several mutation operators for MOTHERA system in FORTRAN [13]. In mutation testing, mutant generation has high cost. So far, several techniques have been proposed in order to reducing the cost. One of the techniques is *Bytecode Translation Technique*. The technique was first proposed by Ma [19, 39]. As a result, Bytecode easily generates mutants in Java from the compiled code, instead of source code. At first, it was thought that most programmers are competent and their programs have several simple faults. Thus, mutation operators that were proposed were very simple and just focused on simple changes of code syntax. You can see some of the mutation operators in [12]. In addition to FORTRAN, Offutt et al. proposed 65 different mutation operators in ADA. Generally, the operators can be divided into five groups: *Operand Replacement Operators*, *Statement Operators*, *Expression Operators*, *Coverage Operators*, and *Tasking Operators*. Agrawal et al. [16] also proposed several different mutation operators in C and compared them with operators of Offutt and Way. According to their result, their operators were able to achieve 99/6% mean mutation score. Kim et al. [17] designed 15 mutation operators for class mutation. The operators can be divided into four categories: *Polymorphic Operators*, *Method Overloading Operators*, *Information Hiding Operators*, and *Exception Handling Operators*. Chevalley [18] also did a work similar to [17]. Derezińska [20] proposed several mutation operators in C# and implemented them as a tool named *Cream* [36].

2.2 Test-Case Generation

At first, process of test-case generation was manually done. It imposed high cost into testing process. Thus, many researchers decided to solve the problem. One of the initial attempts for automatic generation of test-case was use of random algorithm. For e.g. Chen et al. [40], Pacheco et al. [41], and Ciupa et al. [42] used random techniques. In the techniques, they used random algorithm. Offutt was another person that was able to

mmeybodi@aut.ac.ir

develop the research field [43]. He [7] introduced an automatic method for test-case generation in his doctoral thesis, called *Constraint-based Test-Case Generation (CBT)*. Under CBT, a test-case is able to kill a mutant when it satisfies three conditions: *Reachability*, *Necessity* and *Sufficiency*. Offutt and DeMillo [8] implemented a tool for test-case generation in mutation testing named *Godzilla*. Godzilla used CBT and worked on MOTHRA system. In addition to CBT, they implemented Godzilla base on *Control Flow Analysis* and *Symbolic Evaluation*. Their practical result showed that 90% generated mutants can be killed by CBT-based Godzilla. Some researchers were interested to use ET approach for test-case generation. For e.g. Baudry et al. [10] adapted GA and BA for this purpose in C#. As we all know, each EA, depending problem, needs an appropriate fitness function. As a result, they used *Mutation Score* function as fitness function. Ayari et al. [11] also adapted ant-colony algorithm and compared it with HC and GA in Java. Dynamic Symbolic Execution (DSE) is another technique of test-case generation. DSE collects branch predicates of a path. Then, it iteratively attempts to generate test-case that is able to satisfy the predicates. Main criteria in DSE is code coverage [37, 38]. Zhang et al. [23] proposed a new approach in order to generating test-cases that are able to achieve high killing rate. The approach was named PexMutator and worked in C#. PexMutator first translates a program into a meta-program using a set of rules. Then, it attempts to kill mutants using DSE technique. According to its practical result, PexMutator was able to strongly kill 80% the generated mutants. Harman et al. [25] introduced SHOM architecture. SHOM combines DSE and ET techniques for high-quality test-case generation. They carried out their empirical study on 17 different programs. Based on their result, test-cases generated by SHOM were able to achieve high killing rate. Moreover, Harman et al. [26] also examined relation between search space size and performance of ET. In fact, they investigated impact of removing irrelative variables on test-case generation. Fraser and Zeller [24] generated several mutants from class and used ET to kill them. Papadakis et al. [22] implemented a framework in Java, instead of designing a new tool for test-case generation. The framework uses three existing tools: *JPF-SE*, *Concolic* and *Etos*.

Villa et al. [21] proposed two mutation operators for dynamic and static memory allocation. The main goal of the operators is detection of Buffer Overflows (BOF). Tuya et al. [28] introduced several mutation operators for SQL query statement. The operators can be divided into four groups: *SQL Clauses*, *Expressions*, *Handling Null values*, and *identifiers*. They also implemented them as a tool named *SQLMutation.* Hierons and Merayo [29] proposed seven mutation operators for *Finite State Machines*. Zhan and Clark [27] implemented mutation testing system in *MATLAB*. Wang and Huang [31] also applied mutation testing in *web services*. Vigna et al. [30] used mutation testing in order to *detecting malicious Traffic.*

3 Mutation Testing

Mutation testing is a powerful method for software testing, which generally consists of four different units [32] which are displayed in Fig. 1.

mmeybodi@aut.ac.ir

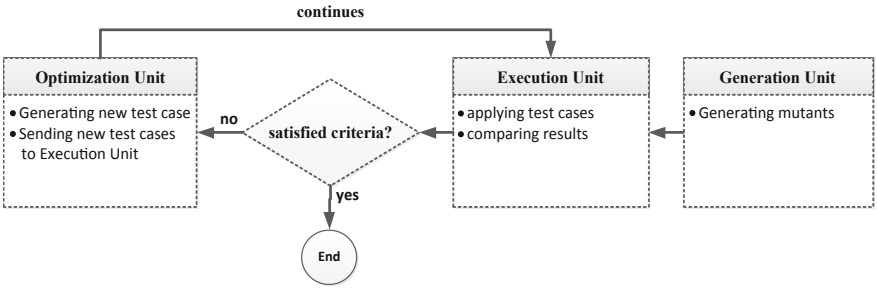


Fig. 1. Mutation testing process

Generation Unit: The unit creates different copies of main program. Next, it injects various faults within the copies using different mutation operators. These copies are called *mutant*. Mutation operators play a key role in mutation testing. In other words, if the mutation operators are designed precisely, output test-cases of mutation testing will have higher potential for detecting faults of a program under test. The simplest mutation operator that can be used is Arithmetic Operator Replacement (AOR). Suppose that (a = b * c) is the main statement to follow. Accordingly, (a = b/c) and (a = b + c) can be generated by AOR. Each mutant can use mutation operators for *n* times, which it is called *n-order Mutant*.

Execution Unit: After generating mutants, execution unit applies the provided test-cases on main program and mutants, and then compares their results with each other. Generally, the results can be compared in two forms: *weak mutation* and *strong mutation*. Strong mutation compares final outputs of main program and mutant. Fully execution of mutants has high computational cost, but weak mutation has partly solved this problem. Weak mutation prevents full execution of mutants and compares internal states of main program and mutants. Anyway, if there is any difference in the results (main program and mutant), it is said that the given test-case has been able to detect the injected fault. As a result, the mutant is called *killed* mutant. Otherwise, the mutant is considered as *live*. If neither of test-cases is able to kill the live mutant, it is called *Equivalent* mutant.

Criterion Unit: Each testing process should continue until reaching a specific criterion. So far, different criterions are proposed for software testing. For e.g., *code coverage*, *path coverage*, *node coverage*, and etc. But one of the useful criterion for mutation testing is *killing mutant count*. In the criterion, main goal is generation of test-cases that is able to kill all or maximum mutants. Readers who are interested to study more about testing criteria can refer to [32].

Optimization Unit: If neither of test-cases is able to satisfy the given criteria, the optimization unit will attempt to generate new test-cases. There are various techniques for test-case generation in mutation testing. One of the useful techniques is the use of heuristic approaches. Heuristic approach works base on ET. ET attempts to generate optimal test-cases using different EAs such as GA, HC and etc. Main goal of mutation testing is generation of test-cases that is able to detect all injected faults. Whatever a

mmeybodi@aut.ac.ir

test-case is able to kill more mutants, it will be a more suitable option for testing a software because it is likely able to detect many faults. Offutt [2] proved coupling effect in one of his researches. According to coupling effect, if a test-case is able to kill 1-order mutants, more likely it will kill n + 1-order mutants. Thus, many researchers usually use 1-order mutants for experimental study. Like the researchers, we also used 1-order mutants. To clarify concept of above definitions, an example of killing mutant is presented. In Fig. 2(a), there is a main program (Find_Max) in which receives three inputs and returns maximum of them. Suppose we have already used the Generation Unit and have generated four mutants. Information of the mutants is in Fig. 2(b). After applying the Generation Unit, it is time to run the Execution Unit. Details of the unit is shown in Fig. 2(c).

Execution Unit							
Decision		comparison of results					
Strong killing	weak killing	Strong results		Weak results		Test-Case	
		Main program	Mutant	Main program	Mutant		
✓	×	30	20	30	30	a=10 , b=20 , c=30	**Mutant 1**
×	✓	8	8	4	3	a=4 , b=7 , c=8	**Mutant 2**
✓	✓	6	5	true	false	a=4 , b=5 , c=6	**Mutant 3**
×	×	3	3	2	2	a=1 ,b=2 , c=3	**Mutant 4**

(c) Execution unit details

```
1   public int Find_MAX(int a, int b, int c)
2   {
3      int max=0;
4      if(a>b)
5         max=a;
6      else
7         max=b;
8      if(c>max)
9         max=c;
10     return max;
11  }
```

(a) Main program

Generation Unit		
line number of replacement	**Mutated statement**	**Mutant ID**
9	c=c;	1
5	max=a-1;	2
8	if(--c>max)	3
7	max=b-- ;	4

(b) 1-order Mutants Information

Fig. 2. An example of killing mutant

As shown, the *Test_Case* column shows test input data of each mutant. The *Weak_Results* column refers to results of main and mutated statement, whereas the *Strong_Results* column refers to the final results of main program and mutant. The *Decision* column also shows type of killing mutants. Other words, each test-case that have been able to strongly or weakly kill the mutants is shown by the column.

Now you consider test input data of mutants (*Test_Case* column). If you look at test input data of mutant 3, you will understand the test-case was able to strongly and weakly kill mutant 3 (marked by ✓). Generation of test-cases that is able to kill a mutant in both weak and strong mutations is preferable. The issue is a research topic and few studies has been done in the field so far. Anyway, now you consider test input data of mutant 4. As is evident, the test-case have not adequate quality because it was able to kill mutant 4 neither weak mutation nor strong mutation (marked by ×). As result, mutant 4 is considered as a live mutant. Since test input data of mutant 4 was not able to detect its injected faults, the test-case should be optimized by the Optimization Unit. One techniques of improving test-case is the use of ET.

mmeybodi@aut.ac.ir

4 Test-Case Generation Based on Evolutionary Testing

The ET is a useful approach for test-case generation. It searches the search space using EAs to generate high-quality test-cases. Since implementation of the paper is based on ET approach, Subsect. 4.1 explains the used EAs. As you know, execution of EAs depends on fitness functions. Accordingly, Subsect. 4.2 also describes the used fitness functions (Fig. 3).

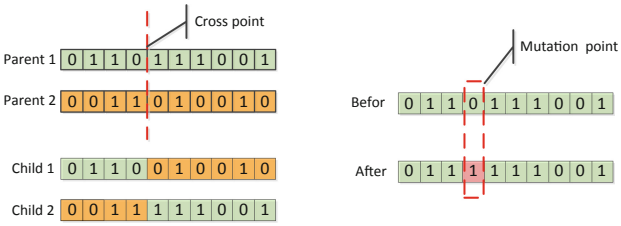


Fig. 3. Crossover and mutation operators

4.1 Evolutionary Algorithms (EAs)

The paper used GA, BA, HC, GA and PSO for test-case generation. In the following, overall process of each the EA is described.

4.1.1. Genetic Algorithm (GA)

GA is derived from genetic science and it is based on two main concepts: *Chromosome* and *Gene*. Figure 4 shows its overall process [10].

1	**Select an initial generation**
2	Evaluation
3	Crossover
4	Mutation on some chromosomes
5	Satisfies the desired conditions

Fig. 4. GA steps

- **Step 1:** in the step, initial test-cases (initial generation) are selected for start of process.
- **Step 2:** step 2 applies the selected test-cases on main program and generated mutants. According to obtained results (mutant and main program), the given fitness function assesses test-cases.
- **Step 3, 4:** the steps apply crossover and mutation operators on test-cases in order to generating new test-cases (new generation). The operators are shown in Fig. 1.
- **Step 5:** step 5 also checks final condition to terminate GA process. The conditions can be considered as achieving to a specific value of fitness function, achieving to

mmeybodi@aut.ac.ir

specific killing rate, generating *n* generations, or etc. Step 2 to 5 continues until the final condition is satisfied by the generated test-cases.

4.1.2. Bacteriological Algorithm (BA)

BA has inspired from bacteria behavior in the nature. Unlike GA, BA only uses mutation operator. Figure 5 shows its overall process [10].

- **Step 1, 2:** the steps are similar to GA. In the steps, initial test-cases are first selected by testers. Then, they are assessed by fitness function.
- **Step 3:** are mutated by mutation operator, test-cases that were not able to achieve a good fitness value. Otherwise, step 3 passes them to the next generation as good test-cases.
- **Step 4:** as GA, termination conditions are checked by the step. Step 2 to 4 continues until the final condition is satisfied by the generated test-cases.

1	**Select an initial generation**
2	Evaluation
3	Keeping and Mutating
4	Satisfies the desired conditions

Fig. 5. BA steps

4.1.3. HC

HC is the most famous and the simplest EA in test-case generation. The key point in HC is that it locally searches the search space. Figure 6 shows its overall process.

1	**Select an initial generation**
2	Evaluation
3	Selecting the best test-case
4	Finding neighbors of the best test-case
5	Satisfies the desired conditions

Fig. 6. HC steps

- **Step 1, 2:** the steps are the same with GA and BA.
- **Step 3:** is selected as the best test-case by the step, test-case that has earned the highest fitness value.
- **Step 4:** the step searches neighbor test-cases of the best test-case. For e.g., suppose [77, 66, 25] is the best test-case in step 3. Thus, test-cases [77, 68, 25] and [77, 66, 20] can be generated as neighbor test-cases by adding or subtracting a specific value.
- **Step 5:** as GA and BA, the step checks termination condition. Step 2 to 5 continues until the final condition is satisfied by the generated test-cases.

mmeybodi@aut.ac.ir

4.1.4. Queen Algorithm with GA Approach

QA emulates bee queen behavior. It is similar to GA except that the best test-case, which has earned the highest fitness value, is combined with all test-cases by crossover operator. Overall process of QA is shown in Fig. 7 [33].

1	**Select an initial generation**
2	Evaluation
3	Selecting the Queen
4	Crossover
5	Mutation on some Bees
6	Satisfies the desired conditions

Fig. 7. QA steps

- **Step 1, 2:** steps 1 and 2 are similar to previous algorithms.
- **Step 3:** is chosen as queen by the step, test-case that has been able to obtain the highest fitness value.
- **Step 4, 5:** steps 4 and 5 applies crossover and mutation operators on test-cases. Step 4 combines the best test-case (queen) with all test-cases using crossover operator. For diversity in generation, step 5 also applies mutation operator on some test-cases.
- **Step 6:** as above, the step checks termination condition. Step 2 to 5 continues until the final condition is satisfied by the generated test-cases.

4.1.5. PSO

PSO is modeled from bird's behavior. It was proposed in 1995 by Eberhart and Kennedy [34]. Figure 8 shows its overall process.

1	**Select an initial generation**
2	Evaluation
3	Gbest replacement
4	Pbest replacement
5	Compute velocity
6	Generate test-case
7	Satisfies the desired conditions

Fig. 8. PSO steps

- **Step 1, 2:** the steps are the same with previous algorithms.
- **Step 3, 4:** PSO composed of two main parameters: *Gbest* and *Pbest*. The best test-case up to current time is kept by *Gbest,* whereas *Pbest* keeps the best test-case in current generation.

mmeybodi@aut.ac.ir

- **Step 5:** Velocity function determines movement speed in the search space. Other words, it specifies change amount of test-cases in order to generating high-quality test-cases. The function is calculated for all test-cases of a generation.

$$V_i(t+1) = wV_i(t) + c_1 r_1[P(t) - T_i(t)] + c_2 r_2[G(t) - T_i(t)]$$

$V_i(t)$ is velocity of i test-case in t time. w, c_1 and c_2 are user coefficients. They should respectively be in ($0 \leq w \leq 1.2$), ($0 \leq c_1 \leq 2$) and ($0 \leq c_2 \leq 2$). r_1 and r_2 are randomly determined in ($0 \leq r_1 \leq 1$, $0 \leq r_2 \leq 1$). $P(t)$ and $G(t)$ are the same *Pbest* and *Gbest* in steps 3 and 4. $T_i(t)$ is i test-case in t time.

- **Step 6:** new test-cases, regarding the computed velocity in step 5, are generated by the step as follows:

$$T_i(t+1) = T_i(t) + V_i(t+1)$$

- **Step 7:** as above, termination condition is checked. Step 2 to 7 continues until the final condition is satisfied by the generated test-cases.

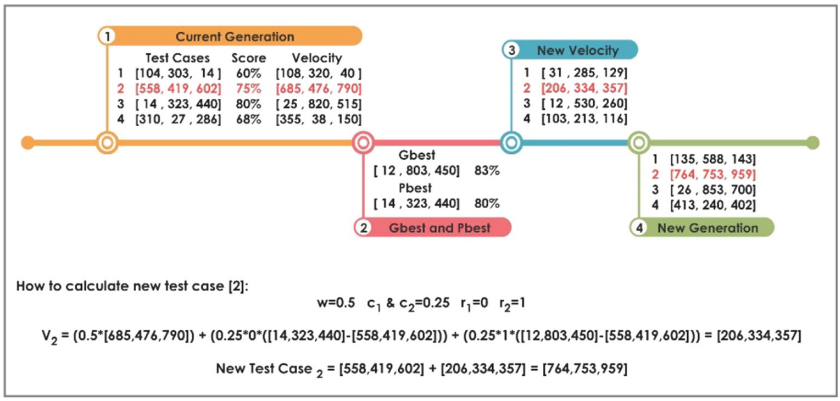


Fig. 9. An example of PSO

An example of test-case generation using PSO is presented in this part to clarify the process. As can be seen in Fig. 9, there is an initial generation. Suppose we want to calculate test-case (2) of the next generation. At first, we should update values of Gbest and Pbest. According to above definitions, Pbest selects test-case (4) as the best data (obtaining 76% fitness value) in current generation, whereas Gbest remains unchanged because no test-cases of current generation was able to obtain higher fitness value than Gbest. Next, velocity of each test-case should be calculated by $V_i(t + 1)$.

mmeybodi@aut.ac.ir

4.2 Fitness Functions

As noted, this subsection explains the used fitness functions throught the paper.

4.2.1. Fitness Function 1

The first function is *Mutation Score (MS)*. It is composed of three main parameters: *E*, *K* and *all* [10].

$$MS = \left(\frac{K}{All - \mathrm{E}}\right) \times 100$$

The *all* keeps total number of the generated mutants. The *k* and *E* respectively refer to number of the killed and equivalent mutants. Generally, whatever a test-case is able to kill more mutants, MS will assigns it a higher score.

4.2.2. Fitness Function 2

The second function is *Branch (BR)* [25]. It works base on branches of a program. BR assigns the highest fitness value to test-case that has been able to make the most different in the satisfied branches between main program and mutants.

$$d(p,m,i,t) = \left\{ \begin{array}{ll} 1 & if\ Branch\,(p,i,t) \neq Branch\,(m,i,t) \\ 0 & if\ Branch\,(p,i,t) = Branch\,(m,i,t) \end{array} \right\}$$

$$BR(p,m,t) = \frac{\sum_{i \in all\ critical\ point} d(p,m,i,t)}{N}$$

Branch(p, i, t) refer to *i* branch in *p* program, which is satisfied by *t* test-case. Similarly, *Branch(m, i, t)* also refer to *i* branch in *m* mutant. *BR(p, m, t)* calculates fitness value of a test-case base on all its different satisfied branches than *n* mutant. Generally, the main goal of BR is that diverges execution path of a test-case between main program and mutants.

4.2.3. Fitness Function 3

The third function is *APP*. It consists of two main parameters: *Approach_level* and *Branch_distance* [26, 35].

$$APP = Approach_Level + Norm\,(Branch_distance)$$

Approach_level is number of nested branches that a test-case should satisfy to reach the mutated statement (*infection point*). *Branch_distance* also refers to a test-case how to close to satisfy the given branch predicate. It should be noted that *Branch_distance* value is normalized as follows:

$$w(x) = 1 - \alpha^{-x}$$

4.2.4. Fitness Function 4

The fourth function is *Result_DIFference_Fitness* (*RDIFF*). As mentioned, the main goal of APP in previous subsection is guidance of test-cases toward the mutated statement (infection point). If you look at the function, you will find that APP has not considered results of main and the mutated statements. Other words, a test-case that has

mmeybodi@aut.ac.ir

reached to the mutated statement may not be able to generate a different result between main and mutated statements. In fact, whatever the results (main and mutated statement) are more different, probability of killing mutants will be higher. According to the point, the paper has tried to cover the issue by adding *Result_Difference* parameter.

$$R_Diff = |P_state_i - M_state_i|$$

$$RDIFF = Approach_Level + Norm(Branch_distance + R_Diff)$$

P-staet$_i$ and *M-state*$_i$ respectively refer to result of i statement in main program and mutant. *R_Diff* also computes difference of the parameters. Generally, whatever a test-case is able to reach the mutated statement and to generate more different result; it will earn higher fitness value.

5 Simulation Results

This section is composed of two subsections: Experimental setup and simulation results. The first subsection presents implementation details and the second subsection displays implementation results using five different tables.

5.1 Experimental Setup

The mutation testing system is implemented using C# language and SQL Server 2008. C# and SQL Server is applied for creating mutation testing system engine and saving the generated results. Since execution of mutation testing depends on mutants, we generated mutants from seven programs which can be seen in Table 1. In the table, specifications such as line count, branch statement count and the generated mutant count of each program are shown. According to the table, The *Trian* is the most famous program in software testing. It receives sides of a triangle as input and detects the type: *Equilateral*, *Isosceles* and etc. The *NextDate* gets a specific date and computes the next date. The *ColorRang* gets three inputs as RGB and computes a 32-bit color spectrum. The *DayFind* receives a specific date and returns weekday. The *MAZE* is a famous problem and finds optimal path among maze paths. The *Zip* is a data compression algorithm. The *Intersec* receives start and end points of several lines as input and returns number of their intersection points as output.

Table 1. Selected programs

Benchmark	Line	Branch	Weak/strong mutants
Trian	65	22	515
NextDate	75	21	313
MAZE	246	32	1737
Intersec	815	140	3637
DayFind	223	39	1970
ColorRang	206	31	424
Zip	489	35	673

mmeybodi@aut.ac.ir

Based on above details, we ran each EAs (GA, QA, BA, HC and PSO) based on fitness functions of 4-2 subsection for 10 times. Then, the mean results are calculated and are shown in the next subsection. The simulation platform used is a Intel core i7 2.9 GHz, RAM: 4 GB on Windows 7 Operating system.

5.2 Simulation Results

Generally, the results are divided into 5 tables. Tables 2 and 3 respectively show weak and strong results of 10 simulation runs. In the tables, T and K columns respectively show average run time and average maximum killed count. K/T also computes ratio of average killed count to average run time.

Table 2. Weak results

		Trian		NextDate		ColorRang		DayFind		Maze		Intersec		Zip		K/T
EA	Fitness	T	K	T	K	T	K	T	K	T	K	T	K	T	K	
GA	MS	592.20	466.10	24.30	160.00	24.40	331.00	17.70	127.90	108.50	1656.00	99.20	2506.00	712.20	852.00	3.86
	RDIFF	3466.00	452.00	10.14	160.00	34.40	324.80	19.90	109.10	91.10	1736.00	222.30	2429.00	73.70	852.00	1.55
	APP	625.90	420.90	12.60	160.00	35.00	332.00	16.20	85.40	34622.00	1736.00	194.10	2412.00	74.20	852.10	0.17
	BR	5.30	444.10	15.80	160.00	34.40	339.20	20.20	103.50	55.60	1736.00	222.80	2452.00	69.10	852.00	14.38
BA	MS	910.50	464.60	16.50	165.00	21.10	380.70	10.90	131.10	17.70	1735.00	59.00	740.40	1023.00	852.00	2.17
	RDIFF	1581.00	477.00	13.23	164.50	35.50	372.80	28.90	112.10	16.70	1734.00	153.00	744.20	50.30	852.00	2.37
	APP	561.90	468.40	14.68	165.00	30.50	380.60	26.20	118.20	32.80	1734.00	168.40	744.70	87.90	852.00	4.84
	BR	9.10	469.20	13.90	165.00	31.80	383.20	20.10	124.40	18.80	1734.00	161.00	740.30	57.40	852.00	14.32
HC	MS	146.70	490.90	17.30	205.10	11.30	234.00	0.30	30.00	4.50	1736.00	64.50	1207.30	300.70	852.00	8.72
	RDIFF	63.80	489.70	9.50	247.30	16.80	249.90	0.30	30.00	11.20	1736.00	182.30	1645.70	27.10	852.00	16.88
	APP	63.80	489.90	14.36	244.10	17.30	243.20	0.30	30.00	12.20	1736.00	166.30	1145.60	32.90	852.00	15.43
	BR	2.10	47.80	11.00	173.10	10.00	278.50	0.30	30.00	10.80	1736.00	152.00	1053.40	135.50	852.90	12.97
QA	MS	4.50	485.50	18.67	160.00	22.70	305.70	18.50	152.00	39.00	1736.00	89.00	2395.20	2582.50	852.20	2.19
	RDIFF	18.60	437.90	14.98	160.00	28.90	331.10	10.60	120.40	46.30	1736.00	374.30	1874.80	263.90	852.00	7.28
	APP	18.50	440.50	13.54	160.00	29.40	339.60	24.50	120.50	178.80	1736.00	226.60	2166.70	146.70	852.00	9.11
	BR	0.10	395.20	12.68	160.00	27.80	309.40	26.00	123.10	22.10	1736.00	223.20	2145.80	281.00	852.00	9.65
PSO	MS	5.40	482.00	15.00	160.10	1.00	284.80	1.50	83.10	5.70	1736.00	64.30	1073.00	260.40	827.80	13.15
	RDIFF	0.00	430.90	13.43	196.20	4.40	319.90	1.80	153.40	9.10	1736.00	174.80	1139.00	40.90	852.00	19.75
	APP	4.60	436.00	12.00	212.10	2.90	330.80	0.60	61.90	6.10	1736.00	169.50	1149.00	48.60	852.00	19.56
	BR	1.30	479.40	13.22	177.10	1.00	265.20	1.90	154.60	4.00	1736.00	66.10	2783.00	51.00	852.00	46.54
RND		171.90	775.40	481.90	22.14	225.10	26.40	391.80	0.30	30.00	11.60	1736.00	59.30	740.20	1644.00	854.10

Table 3. Strong results

		Trian		NextDate		ColorRang		DayFind		Maze		Intersec		Zip		K/T
EA	Fitness	T	K	T	K	T	K	T	K	T	K	T	K	T	K	
GA	MS	592.20	466.10	24.30	160.00	24.40	331.00	17.70	127.90	108.50	1656.00	99.20	2506.00	712.20	852.00	3.86
	RDIFF	3466.00	452.00	10.14	160.00	34.40	324.80	19.90	109.10	91.10	1736.00	222.30	2429.00	73.70	852.00	1.55
	APP	625.90	420.90	12.60	160.00	35.00	332.00	16.20	85.40	34622.00	1736.00	194.10	2412.00	74.20	852.10	0.17
	BR	5.30	444.10	15.80	160.00	34.40	339.20	20.20	103.50	55.60	1736.00	222.80	2452.00	69.10	852.00	14.38
BA	MS	910.50	464.60	16.50	165.00	21.10	380.70	10.90	131.10	17.70	1735.00	59.00	740.40	1023.00	852.00	2.17
	RDIFF	1581.00	477.00	13.23	164.50	35.50	372.80	28.90	112.10	16.70	1734.00	153.00	744.20	50.30	852.00	2.37
	APP	561.90	468.40	14.68	165.00	30.50	380.60	26.20	118.20	32.80	1734.00	168.40	744.70	87.90	852.00	4.84
	BR	9.10	469.20	13.90	165.00	31.80	383.20	20.10	124.40	18.80	1734.00	161.00	740.30	57.40	852.00	14.32
HC	MS	146.70	490.90	17.30	205.10	11.30	234.00	0.30	30.00	4.50	1736.00	64.50	1207.30	300.70	852.00	8.72
	RDIFF	63.80	489.70	9.50	247.30	16.80	249.90	0.30	30.00	11.20	1736.00	182.30	1645.70	27.10	852.00	16.88
	APP	63.80	489.90	14.36	244.10	17.30	243.20	0.30	30.00	12.20	1736.00	166.30	1145.60	32.90	852.00	15.43
	BR	2.10	47.80	11.00	173.10	10.00	278.50	0.30	30.00	10.80	1736.00	152.00	1053.40	135.50	852.90	12.97
QA	MS	4.50	485.50	18.67	160.00	22.70	305.70	18.50	152.00	39.00	1736.00	89.00	2395.20	2582.50	852.20	2.19
	RDIFF	18.60	437.90	14.98	160.00	28.90	331.10	10.60	120.40	46.30	1736.00	374.30	1874.80	263.90	852.00	7.28
	APP	18.50	440.50	13.54	160.00	29.40	339.60	24.50	120.50	178.80	1736.00	226.60	2166.70	146.70	852.00	9.11
	BR	0.10	395.20	12.68	160.00	27.80	309.40	26.00	123.10	22.10	1736.00	223.20	2145.80	281.00	852.00	9.65
PSO	MS	5.40	482.00	15.00	160.10	1.00	284.80	1.50	83.10	5.70	1736.00	64.30	1073.00	260.40	827.80	13.15
	RDIFF	0.00	430.90	13.43	196.20	4.40	319.90	1.80	153.40	9.10	1736.00	174.80	1139.00	40.90	852.00	19.75
	APP	4.60	436.00	12.00	212.10	2.90	330.80	0.60	61.90	6.10	1736.00	169.50	1149.00	48.60	852.00	19.56
	BR	1.30	479.40	13.22	177.10	1.00	265.20	1.90	154.60	4.00	1736.00	66.10	2783.00	51.00	852.00	46.54
RND		775.40	775.40	481.90	22.14	225.10	26.40	391.80	0.30	30.00	11.60	1736.00	59.30	740.20	1644.00	854.10

One of goals to pursue in mutation testing is apply a technique to able to kill all or maximum mutants in both weak and strong mutations. Thus, Table 4 compares EAs and fitness functions that have been able to strongly and weakly kill maximum mutants.

mmeybodi@aut.ac.ir

Table 4. Maximum weakly killed mutants VS. Maximum strongly killed mutants

	Weak			Strong		
	EA	**Fitness**	**Max. Killed**	**EA**	**Fitness**	**Max. Killed**
Trian	PSO	BR	402.60	HC	MS	490.90
NextDate	HC	RDIFF	248.60	HC	RDIFF	247.30
ColorRag	RND	-	318.60	RND	-	391.80
DayFind	QA	MS	86.70	PSO	BR	154.60
Maze	PSO	MS	563.50	GA	RDIFF-APP-BR	1736.00
				HC	MS-RDIFF-APP-BR	
				QA	MS-RDIFF-APP-BR	
				PSO	MS-RDIFF-APP-BR	
				RND	-	
Intersec	GA	RDIFF	2575.00	PSO	BR	2783.00
Zip	BA	MS-RDIFF- APP	662.00	RND	-	854.10

In order to having an overall view of performance, Tables 5 and 6 display strong and weak coverage of EAs (GA, QA, BA, HC and PSO) and fitness functions (MS, RDIFF, APP, BR) for all 9269 mutants.

Table 5. Weak coverage

	GA		BA		HC		QA		PSO		AVG
	Time	**Coverage**	**Time**	**Coverage**	**Time**	**Coverage**	**Time**	**Coverage**	**Time**	**Coverage**	**Coverage**
MS	1105.00	46.52%	521.20	30.98%	362.40	32.71%	806.30	46.93%	478.10	34.01%	38.23%
RDIFF	850.00	45.12%	515.00	29.84%	243.90	38.11%	475.00	40.67%	239.70	32.74%	37.29%
APP	475.60	45.57%	354.60	30.39%	260.10	35.77%	539.00	44.17%	252.20	35.04%	38.19%
BR	478.50	45.24%	227.10	29.49%	217.00	30.87%	494.60	41.59%	143.10	46.41%	38.72%
AVG	727.28	45.61%	404.48	30.18%	270.85	34.36%	578.73	43.34%	278.28	37.05%	

RND	**Time**	**Coverage**
	518.00	27.61%

Table 6. Strong coverage

	GA		BA		HC		QA		PSO		AVG
	Time	**Coverage**	**Time**	**Coverage**	**Time**	**Coverage**	**Time**	**Coverage**	**Time**	**Coverage**	**Coverage**
MS	1554.20	66.06%	2042.20	48.40%	528.00	51.50%	2756.20	65.92%	338.30	50.33%	56.44%
RDIFF	3907.40	65.67%	1865.40	48.27%	301.50	56.87%	742.60	59.70%	231.00	52.28%	56.56%
APP	35567.40	64.97%	907.70	48.34%	292.80	51.35%	624.50	62.98%	232.30	51.75%	55.88%
BR	407.40	65.92%	298.20	48.39%	310.70	45.18%	580.20	61.97%	125.30	69.83%	58.26%
AVG	10359.10	65.65%	1278.38	48.35%	358.25	51.22%	1175.88	62.64%	231.73	56.05%	

RND	**Time**	**Coverage**
	2517.20	48.30%

As mentioned above, the paper used GA, BA, QA, HC, and PSO. Regarding the point, now you consider Figs. 5 and 6. As evident, GA, QA and BA have been able to achieve the highest coverage rate in both weak and strong mutation with a common fitness function (MS). Other words, It can be inferred that MS has been a suitable fitness function for the algorithms. But since PSO and HC have different structure, different fitness functions have guided them. For e.g., PSO using IF and HC using RDIFF have been able to achieve the highest coverage rate (the states are shown with different color in Tables 5 and 6). One problem of testers in using EAs is that they do not know which fitness function is appropriate for guiding in the search space. As a

mmeybodi@aut.ac.ir

result, Tables 5 and 6 can draw a road map for testers that are interested to use EAs. Another point that should be noted is the proposed fitness function of the paper (RDIFF). As explained above, RDIFF is the edited version of APP function and the only its difference with APP is that it has an extra parameter. Regarding the description, as can be seen from Figs. 5 and 6 none of the algorithms has been able to achieve the highest coverage rate using APP. Other words, it can be inferred that RDIFF have had better performance than APP. Of course, it should be noted that results of the paper has obtained in limited conditions and requires more study.

6 Conclusion

One goals of test-case generation in mutation testing is to apply techniques that can be able to strongly and weakly kill all or maximum mutants. As a result of the work no identical fitness function and algorithm was able to achieve the highest killing rate in weak and strong mutations. QA with MS function has been able to weakly kill maximum mutants (86), whereas PSO using BR function has been succeeded to strongly kill maximum mutants (154). Since the paper has evaluated its results based on 1-order mutants, implementation conditions can be extended by adding 2-order or n-order mutants. Moreover, other fitness functions and EAs can be evaluated.

References

1. Lipton, R.: "Fault Diagnosis of Computer Programs" student report, Carnegie Mellon University (1971)
2. DeMillo, R.A., Lipton, R.J., Sayward, F.G.: Hints on test data selection: help for the practicing programmer. Computer **11**(4), 34–41 (1978)
3. Hamlet, R.G.: Testing programs with the aid of a compiler. IEEE Trans. Softw. Eng. **3**(4), 279–290 (1977)
4. Walsh, P.J.: A measure of test completeness. Ph.D. thesis, State University of New York at Binghamton (1985)
5. Frankl, P.G., Weiss, S.N., Hu, C.: All-uses vs. mutation testing: An experimental comparison of effectiveness. J. Syst. Softw. **38**(3), 235–253 (1997)
6. Offutt, J., Pan, J., Tewary, K., Zhang, T.: An experimental evaluation of data flow and mutation testing. Softw.: Practice Exp. **26**(2), 165–176 (1996)
7. Offutt, A.J.: Automatic test data generation. Ph.D. thesis, Georgia Institute of Technology (1988)
8. DeMillo, R.A., Offutt, A.J.: Constraint-based automatic test data generation. IEEE Trans. Softw. Eng. **17**(9), 900–910 (1991)
9. Offutt, A.J., Jin, Z., Pan, J.: The dynamic domain reduction approach for test data generation: design and algorithms. Technical report ISSE-TR-94-110, George Mason University (1994)
10. Baudry, B., Fleurey, F., Jezequel, J.-M., Le Traon, Y.: Genes and bacteria for automatic test-cases optimization in the .NET environment. In: Proceedings of 13th International Symposium Software Reliability Engineering, pp. 195–206 (2002)
11. Ayari, K., Bouktif, S., Antoniol, G.: Automatic mutation test input data generation via ant colony. In: Proceedings of Genetic and Evolutionary Computation Conference, pp. 1074–1081 (2007)

mmeybodi@aut.ac.ir

12. Acree, A.T., Budd, T.A., DeMillo, R.A., Lipton, R.J., Sayward, F.G.: Mutation analysis. Technical report GIT-ICS-79/08, Georgia Institute of Technology (1979)
13. King, K.N., Offutt, A.J.: A Fortran language system for mutation-based software testing. Softw.: Practice Exp. **21**(7), 685–718 (1991)
14. Offutt, A.J., King, K.N.: A Fortran 77 interpreter for mutation analysis. ACM SIGPLAN Not. **22**(7), 177–188 (1987)
15. Offutt, A.J., Voas, J., Payn, J.: Mutation operators for Ada. Technical report ISSE-TR-96-09, George Mason University (1996)
16. Agrawal, H., DeMillo, R.A., Hathaway, B., Hsu, W., Krauser, E.W., Martin, R.J., Mathur, A.P., Spafford, E.: Design of mutant operators for the C programming language. Technical report SERC-TR-41-P, Purdue University (1989)
17. Kim, S., Clark, J.A., McDermid, J.A.: Investigating the effectiveness of object-oriented testing strategies using the mutation method. In: Proceedings of First Workshop Mutation Analysis, pp. 207–225 (2000)
18. Chevalley, P.: Applying mutation analysis for object-oriented programs using a reflective approach. In: Proceedings of Eighth Asia-Pacific Software Engineering Conference, p. 267 (2001)
19. Ma, Y.S., Offutt, A.J., Kwon, Y.-R.: MuJava: an automated class mutation system. Softw. Testing Verif. Reliab. **15**(2), 97–133 (2005)
20. Derezińska, A.: Advanced mutation operators applicable in C# programs. Technical report, Warsaw University of Technology (2005)
21. Vilela, P., Machado, M., Wong, W.E.: Testing for security vulnerabilities in software. In: Proceedings of Conference Software Engineering and Applications (2002)
22. Papadakis, M., Malevris, N., Kallia, M.: Towards automating the generation of mutation tests. In: Proceedings of the 5th Workshop on Automation of Software Test, Cape Town, South Africa, pp. 111–118 (2010)
23. Zhang, L., Xie, T., Zhang, L., Tillmann, N., Halleux, J., Mei, H.: Test generation via dynamic symbolic execution for mutation testing. In: Proceeding of IEEE International Conference on Software Maintenance, Timisoara, Romania, pp. 1–10 (2010)
24. Fraser, G., Zeller, A.: Mutation-driven generation of unit tests and oracles. IEEE Trans. Softw. Eng. **38**(2), 278–292 (2012)
25. Harman, M., Jia, Y., Langdon, W.B.: Strong higher order mutation-based test data generation. In: Proceedings of Conference the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of software engineering, Szeged, Hungary (2011)
26. Harman, M., Hassoun, Y., Lakhotia, K., McMinn, P., Wegener, J.: The impact of input domain reduction on search-based test data generation. In: Proceedings of 6th Joint Meeting European Software Engineering Conference ACM SIGSOFT Symposium Foundations Software Engineering, pp. 155–164 (2007)
27. Zhan, Y., Clark, J.A.: Search-based mutation testing for simulink models. In: Proceedings Conference Genetic and Evolutionary Computation, pp. 1061–1068 (2005)
28. Tuya, J., Cabal, M.J.S., de la Riva, C.: SQLMutation: a tool to generate mutants of SQL database queries. In: Proceedings of Second Workshop Mutation Analysis, p. 1 (2006)
29. Hierons, R.M., Merayo, M.G.: Mutation testing from probabilistic finite state machines. In: Proceedings of Third Workshop Mutation Analysis, published with Proceedings Second Testing: Academic and Industrial Conference Practice and Research Techniques, pp. 141–150 (2007)
30. Vigna, G., Robertson, W., Balzarotti, D.: Testing network-based intrusion detection signatures using mutant exploits. In: Proceedings of 11th ACM Conference Computer and Communication Security, pp. 21–30 (2004)

mmeybodi@aut.ac.ir

31. Wang, R., Huang, N.: Requirement model-based mutation testing for web service. In: Proceedings of Fourth International Conference Next Generation Web Services Practices, pp. 71–76 (2008)
32. Ammann, P., Offutt, J.: Introduction to Software Testing. Cambridge University Press, Cambridge (2008)
33. Qin, L.D., Jiang, Q.Y., Zou, Z.Y., Cao, Y.J.: A queen-bee evolution based on genetic algorithm for economic power dispatch. In: Proceedings of Conference UPEC 2004. 39th International, vol. 1, pp. 453–456 (2004)
34. van den Bergh, F.: An analysis of particle swarm optimizers. Ph.D. thesis, University of Pretoria (2002)
35. Wegener, J., Baresel, A., Sthamer, H.: Evolutionary test environment for automatic structural testing. Inf. Softw. Technol. **43**(14), 841–854 (2001)
36. Derezinska, A., Szustek, A.: CREAM—a system for object-oriented mutation of C# programs. Technical report, Warsaw University of Technology (2007)
37. Godefroid, P., Klarlund, N., Sen, K.: DART: directed automated random testing. In: Proceedings of the 2005 ACM SIGPLAN Conference Programming Language Design and Implementation (PLDI 2005), Chicago, Illinois, USA, 11–15 June 2005, vol. 40, pp. 213–223. ACM (2005)
38. Sen, K., Marinov, D., Agha, G.: CUTE: a concolic unit testing engine for C. In: Proceedings of 13th ACM SIGSOFT International Symposium Foundations of Software Engineering, pp. 263–272 (2005)
39. Offutt, A.J., Ma, Y.-S., Kwon, Y.-R.: An experimental mutation system for Java. ACM SIGSOFT Softw. Eng. Notes **29**(5), 1–4 (2004)
40. Chen, T., Merkel, R., Wong, P., Eddy, G.: Adaptive random testing through dynamic partitioning. In: Fourth International Conference on Quality Software, pp. 79–86 (2004)
41. Pacheco, C., Lahiri, S.K., Ernst, M.D., Ball, T.: Feedback-directed random test generation. In: Proceedings of the 29th International Conference on Software Engineering, pp. 75–84 (2007)
42. Ciupa, I., Leitner, A., Oriol, M., Meyer, B.: ARTOO: adaptive random testing for object-oriented software. In: Proceedings of the 30th International Conference on Software Engineering, pp. 71–80 (2008)
43. Farzaneh, H., Bakhshayeshi, S., Ebrahimi Atani, R.: A survey on test data generation techniques based on Mutation Testing. Soft Comput. J. **2**(1), 72–85 (2013)

mmeybodi@aut.ac.ir

Density Clustering Based Data Association Approach for Tracking Multiple Targets in Cluttered Environment

Mousa Nazari and Saeid Pashazadeh(✉)

Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran
pashazadeh@tabrizu.ac.ir

Abstract. Tracking of multiple targets in heavy cluttered environments is a big challenge. One usual approach to overcome this problem is using data association process. In this study, a novel fuzzy data association based on density clustering for multi-target tracking is proposed. In the proposed algorithm, the density clustering approach is used to cluster the measured data points. This approach is used instead of gates to eliminate false alarms that originate from invalid measurements. Then the association weights of the validated measurements are determined based on the maximum entropy fuzzy clustering principle. The efficiency and effectiveness of the proposed algorithm are compared with JPDAF, MEF-JPDAF and Fuzzy-GA. The results demonstrate the main advantages of the proposed algorithm, such as its simplicity and suitability for real-time applications in cluttered environments.

Keywords: Data association · Fuzzy density clustering · Multi-target tracking · Cluttered environments

1 Introduction

Target state estimation and prediction are the main objectives of tracking systems. The performance of multi-target tracking systems is dependent on two important factors: data association and track filtering. Recursive Bayesian filters e.g. Kalman or particle filter are usually employed as tracking filters and consist of prediction and updating steps. In dense environments, "clutter" or false alarms exist alongside real measurements [1]. The actual measurement origin is unclear and for a measurement cannot be determined that its origin is the targets or environment's clutter. Gating techniques are applied to eliminate false alarms of invalid measurements. Associating valid measurements with existing tracks is done through a data association process. Data association is one of the most essential components of tracking systems in such environments, and it has attracted a lot of attention in the past decades.

A large number of methods to solve data association problems have been proposed [2–5]. Nearest-neighbor based strategies are the simplest data association methods. The nearest measurement of the predicted target position is used to update the target

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 76–88, 2020.
https://doi.org/10.1007/978-3-030-37309-2_7
mmeybodi@aut.ac.ir

trajectory [6]. The Suboptimal Nearest Neighbor (SNN) and Global Nearest Neighbor (GNN) are two prominent nearest-neighbor based strategies.

The Multi-Hypothesis Tracker (MHT) proposed by Donal Reid [7] is the optimal solution for the data association problem in multi-target tracking systems. This method maintains multiple hypotheses that associate past measurements with targets, after which it yields a new set of measurements and calculates the posterior probability using the Bayes rule. Keeping all possible association hypotheses, whereby the number of association hypotheses grows exponentially over time, does not allow this method to be applied in real-time multi-target tracking. Another advanced data association technique is probabilistic data association (PDA). The PDA was proposed by Bar-Shalom and Fortman [5] and is only feasible when one target is available. Based on this approach, joint probabilistic data association (JPDAF) was extended in the case of multi-target tracking. Unlike the nearest-neighbor approach, JPDAF combines validated measurements with different probability association weights rather than selecting a single measurement.

Generally, determining the optimal response for data association has a much computational overhead. Accordingly, the use of soft computing as suboptimal techniques is preferred to complex optimal methods. The soft computing based data association techniques can be grouped as fuzzy logic, neural networks, and evolution algorithm. Fuzzy logic techniques have been proven very successful in performing data association in recent years. For solving the data association problem, two kinds of fuzzy logic technique can be used, including fuzzy inference [8, 9] and fuzzy clustering [10–13]. Osman et al. [14] proposed fuzzy set and fuzzy knowledge-based data association, whereby fuzzy IF-THEN rules are employed in the data association process. The fuzzy knowledge-based approach was first proposed by Singh and Bailey [15] for data association in multi-sensor multi-target tracking. However, increasing the number of targets causes exponential growth of fuzzy rules' number, hence this approach seems inappropriate.

A fuzzy logic association based on fuzzy clustering for solving multi-target data association was developed by Smith [16]. In this approach, the clustering membership degree is used to determine the association weights. The FCM clustering proposed by Bezdek [17, 18] is one of the most well-known and simple algorithms for cluster analysis. This algorithm has often been applied in data association problem research. Nonetheless, FCM may encounter falling into the local minimum. Hence, a fuzzy association based on evolutionary computing for overcoming the local minima problem was developed by Satapathi and Srihari [2]. In this approach, GA and PSO algorithms are used to optimize the distance between cluster centers and the valid measurement data in the FCM. Another soft computing technique for solving multi-target data association is an artificial neural network (ANN) [19]. These categories of soft computing data association techniques have been less considered due to the high number of required neurons.

As mentioned above, the measurement origin is uncertain and is generally not know that it originated from targets or other phenomena. Thus, gating is employed prior to data association to eliminate implausible measurements. Gating is in fact an area in the sensor view where we expect to sense target's measurement(s) effects [20, 21]. Gate size and multiple tracks falling within the gate(s) are practical problems in gate application. A detailed description of gating methods can be found in [11, 21].

mmeybodi@aut.ac.ir

However, data association efficiency is directly dependent on gating results. In this paper, a new fuzzy data association based on density clustering and maximum entropy is proposed for tracking multiple targets in a cluttered environment. Unlike other methods, density clustering facilitates selecting valid measurements. Besides, maximum entropy fuzzy clustering allows calculating the associated probability between valid measurements and tracks.

The remainder of this paper is organized as follows. A brief introduction to density clustering and maximum entropy fuzzy clustering is presented in Sect. 2. Section 3 discusses the basic elements of our proposed method named fuzzy density clustering joint probabilistic data association filter (FD-JPDAF). The simulation results and performance comparisons are presented in Sect. 4 and the conclusions are provided in Sect. 5.

2 Background

2.1 The Stochastic Model

Suppose that there are T targets under surveillance and the dynamics and measurement models of target $j\{j = 1, 2, \ldots, T\}$ are defined respectively as follows:

$$x_j(k) = F_j(k)x_j(k-1) + G_j(k)v_j(k) \tag{1}$$

$$z_j(k) = H_j(k)x_j(k) + w_j(k) \tag{2}$$

where $x_j(k)$ is an n-dimensional state vector, and $z_j(k)$ is an m-dimensional measurement vector of the j^{th} target at time k. $F_j(k)$ is an $n \times n$ state transition matrix, $G_j(k)$ is an $n \times m$ noise matrix, and $H_j(k)$ is an $m \times n$ measurement transition matrix [22]. The process noise $v_j(k)$ and measurement noise $w_j(k)$ are independent zero mean Gaussian noise vectors with known covariance $Q_j(k)$ and $R_j(k)$, respectively.

$$Q_j(k) = Cov\big(v_j(k)\big) \tag{3}$$

$$R_j(k) = Cov\big(w_j(k)\big) \tag{4}$$

If the measurements do not contain any clutter or ECM (noise free environment), the simple Kalman filter is used to predict and update of tracks [23, 24].

$$\hat{x}_j(k+1|k) = F_j\hat{x}_j(k|k) \tag{5}$$

$$P_j(k+1|k) = F_jP_j(k|k)F_j^T + Q_j(k) \tag{6}$$

$$\hat{x}_j(k+1|k+1) = \hat{x}_j(k+1|k) + K_j(k+1)\tilde{z}_j(k+1) \tag{7}$$

$$P_j(k+1|k+1) = \big[\mathrm{I} - K_j(k+1)H_j(k+1)\big]P_j(k+1|k) \tag{8}$$

where $\tilde{z}_j(k)$ is the sum of all weighted innovations and $K_j(k)$ is the Kalman filter gain:

mmeybodi@aut.ac.ir

$$\tilde{z}_j(k) = z_j(k+1) - H_j(k+1)\hat{x}_j(k+1|k) \tag{9}$$

$$K_j(k) = P_j(k|k-1)H_j(k)^T\left[H_j(k)P_j(k|k-1)H_j(k)^T + R_j(k)\right]^{-1} \tag{10}$$

The innovation covariance matrix is given by

$$S_j(k) = H_j(k)P_j(k|k-1)H_j(k)^T + R_j(k) \tag{11}$$

2.2 Density Clustering

Clustering is the process of finding similarities between data points and grouping them into clusters. Over the last decades, numerous clustering algorithms have been proposed, which can be classified into partitioning, hierarchical, density-based and grid-based methods.

Density clustering is a nonparametric approach that the number of clusters is not required as an input parameter and can discover clusters of arbitrary shape and appropriately handles noises [25]. Density Based Spatial Clustering of Applications with Noise (DBSCAN) is the most popular density-based clustering technique, which was proposed by Ester et al. [26–28]. This algorithm requires only two important input parameters, *Eps* (maximum radius of the neighborhood) and *MinPts* (minimum number of points in the cluster). Based on these parameters, dataset points are classified as core points, border points and outliers (noise) as follows:

- Core Point: If at least *MinPts* points are within *Eps* of the point, it is a core point.
- Border Point: a point q is border point if it is not a core point but there is in Eps-neighborhood of a core point.
- Outlier Point: any point that is not a core nor a border point.

Where Eps-neighborhood is set points within *Eps* of the core point (p) $\{ q \in D | dist(p, q) \leq Eps \}$. The algorithm starts with arbitrary point p, and retrieves Eps-neighborhood of point p. If p is a core point, a new cluster is formed and the point p and its Eps-neighborhood are added into this new cluster. To continue, core points and border points of Eps-neighborhood cluster member are added to the cluster. The process is repeated until all the points are either assigned to some cluster or marked noise [29, 30].

2.3 Maximum Entropy Fuzzy Clustering

Suppose $X = \{x_i, i = 1, \ldots, N\}$ be a set of data in R^d and related to one of the clusters $\{c_j, j = 1, \ldots, C\}$. The objective function of clustering process can be defined as follow [22]:

$$E = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} d(x_i, c_j) \tag{12}$$

mmeybodi@aut.ac.ir

Where $d(x_i, c_j)$ is the squared Euclidean distance from data point x_i to the cluster centre c_j, and u_{ij} is the fuzzy membership of x_i to cluster c_j, which satisfies the following conditions:

$$0 \leq u_{ij} \leq 1, \forall i, j \tag{13}$$

$$\sum_{j=1}^{C} u_{ij} = 1, \forall i \tag{14}$$

where membership u_{ij} is determined based on the maximum entropy principle, whereby the Shannon entropy given as follows:

$$H(u_{ij}) = -\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \ln u_{ij} \tag{15}$$

is maximized under the restrictions in (13) and (14). Using the Lagrange multiplier method, the objective function can be defined as

$$\begin{aligned} J(U, C) = &-\sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij} \ln u_{ij} \\ &-\sum_{i=1}^{N} \alpha_i \sum_{j=1}^{C} u_{ij} d(x_i, c_j) + \sum_{i=1}^{N} \lambda_i \left(\sum_{j=1}^{C} u_{ij} - 1 \right) \end{aligned} \tag{16}$$

Finally, the membership degree of x_i to cluster c_j is derived as follows:

$$u_{ij} = \frac{e^{-\alpha_i d(x_i, c_j)}}{\sum_{j=1}^{C} e^{-\alpha_i d(x_i, c_j)}} \tag{17}$$

where α_i and λ_i are Lagrange multipliers. Parameter α_i is known as the "discriminating factor" whose optimal value that was proposed by Liangqun et al. [22] is as follows:

$$\alpha_{opt} = -\frac{\ln \varepsilon}{d_{min}} \tag{18}$$

where d_{min} denotes the distance between x_i and the nearest cluster centre c, i.e., $d_{min} = d(x_i, c) \leq d(x_l, c) for\, l = 1, \ldots, N\, and\, i \neq l$, and ε is a small positive constant. A detailed derivation of the maximum entropy fuzzy clustering can be found in many researches [13, 22, 31, 32].

Maximum entropy fuzzy clustering has become prominent with the advancement in target tracking. This method was first used for robotic tracking by Liu and Meng [31]. A modified version for real-time target tracking applications was later proposed [22]. In order to solve the maneuvring problem of target, Li and Xie [13] proposed the interacting multiple model (IMM) based on maximum entropy fuzzy clustering.

mmeybodi@aut.ac.ir

3 Fuzzy Density Data Association

Despite the excellent performance of fuzzy data association methods, they involve an extra step compared to the non-fuzzy data association methods. Nonetheless, similar to other methods, gating is used to eliminate invalid measurements. The efficiency of fuzzy data association methods is therefore dependent on gates and their characteristics such as gate size, gate type, etc. To overcome this shortcoming, a new fuzzy data association filter without the need for gating is proposed.

Suppose a measurement set $\{z_i, i = 1, \ldots, N_k\}$ is related to target set $\{t_j, j = 1, \ldots, T\}$ at time k. In the first step, the density clustering approach is used to cluster the measurements. The number of clusters is equal to the number of targets and the algorithm considered the points $(\hat{x}_j(k+1|k))$ as the core point to restore Eps-neighborhood. Then, based on MinPts and Eps parameters, Eps-neighborhood of the points $\hat{x}_j(k+1|k)$ are determined as the core points and border points. Core points and border points' measurements were determined as the valid measurements while the outliers were considered as invalid measurements. At the end of clustering process, the predicted target positions removed from the valid measurements.

$$\boldsymbol{\beta} = \{\beta_i^j\} = \begin{bmatrix} \beta_1^1 & \beta_2^1 & \cdots & \beta_{m_k}^1 \\ \beta_1^2 & \beta_2^2 & \cdots & \beta_{m_k}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_1^T & \beta_2^T & \cdots & \beta_{m_k}^T \end{bmatrix} \tag{19}$$

$$\beta_i^j = \begin{cases} u_{ij} & \begin{array}{c} \textit{if the measurement } z_i \textit{ is a valid} \\ \textit{measurement of the target } j. \end{array} \\ 0 & \textit{Otherwise} \end{cases} \tag{20}$$

$$\sum_{i=1}^{m_k} \beta_i^j = 1 \tag{21}$$

where β_i^j is the association probability between measurement z_i and target j, m_k is the number of valid measurements from the previous step, and u_{ij} is the degree of membership of measurement z_i belonging to target j, which is obtained with (17).

Associating one measurement with multiple targets and more than one measurement originating from one true target are problems in highly complex environments. The association probability matrix is reconstructed for measurement(s) associated with multiple targets as follows:

$$\beta_i^j = \begin{cases} \beta_i^j & \text{if } \beta_i^j = max_{l=1:m_k}\beta_l^j, \\ min_{i\in\gamma}\{\beta_l^j\} & otherwise \end{cases} \tag{22}$$

where γ is the set of all tracks associated with measurement z_i. The main idea of this rule is based on the second basic hypothesis of JPDAF [5] that there is only one true measurement originated from each target. So the association probability of the

mmeybodi@aut.ac.ir

measurement j with highest value remains unchanged and the rest of the association probability will be set to the minimum value of the probabilities. Eventually, the modified probability matrix β can be reconstructed as:

$$\bar{\boldsymbol{\beta}} = \begin{bmatrix} \beta_1^1/N_1 & \beta_2^1/N_1 & \cdots & \beta_{m_k}^1/N_1 \\ \beta_1^2/N_2 & \beta_2^1/N_2 & \cdots & \beta_{m_k}^1/N_2 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_1^{\mathrm{T}}/N_T & \beta_2^{\mathrm{T}}/N_T & \cdots & \beta_{m_k}^{\mathrm{T}}/N_T \end{bmatrix}, \tag{23}$$

where $\bar{\beta}$ is normalized association probability matrix, and $N_t = \sum_i^{m_k} \beta_i^t \quad t = 1, \ldots, T$. Steps of Our proposed FD-JPDAF method is briefly summarized in the following steps.

Step 1. $x_j(k-1|k-1)$ and $p_j(k-1|k-1)$ are estimated for each target at k-1 time, $j = 1, \ldots, T$. Then the target state is predicted as follows:

$$x_j(k|k-1) = F_j(k)x_j(k-1|k-1) + G_j(k)v_j(k) \tag{24}$$

$$P_j(k|k-1) = F_j(k)P_j(k-1|k-1)F_j(k)^T + G_j(k)Q_j(k)G_j(k)^T \tag{25}$$

Step 2. The clustering measurement data are set and unlikely measurements are eliminated based on the predicted target positions in the previous step.
Step 3. Membership degree matrix U is computed using (17).
Step 4. Association probability matrix β is computed using (19) and (21), and reconstruction is done as required based on (22) and (23).
Step 5. The target states are updated and the covariance is estimated as:

$$x_j(k|k) = x_j(k|k-1) + K_j(k)\tilde{z}_j(k) \tag{26}$$

$$P_j(k|k) = P_j(k|k-1) - K_j(k) \left[\sum_{i=1}^{m_k} \beta_i^j \tilde{z}_i^j(k)\tilde{z}_i^j(k)^T - \tilde{z}_i^j(k)\tilde{z}_i^j(k)^T \right] - K_j(k) \tag{27}$$

where $K_j(k)$ is the Kalman filter gain (10) and $\tilde{z}_j(k)$ is the sum of all weighted innovations:

$$\tilde{z}_j(k) = \sum_{i=1}^{m_k} \beta_i^j \tilde{z}_i^j(k) \tag{28}$$

Step 6. Steps 1–5 are repeated for the next time step.
According to the description of FD-JPDAF, a simple diagram of a tracking system based on this new approach is presented in Fig. 1. As seen in this diagram, FD-JPDAF does not need to use the gating method and consequently has fewer steps than

mmeybodi@aut.ac.ir

other fuzzy data association methods. It is also expected to be more flexible than other methods owing to the use of the density clustering approach to eliminate invalid measurements.

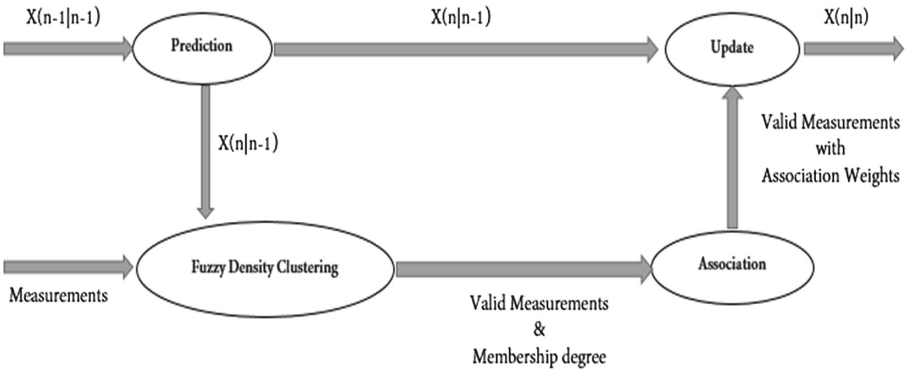


Fig. 1. Simple diagram of tracking system based on fuzzy density data association.

4 Results and Discussion

For a performance comparison and evaluation of FD-JPDAF, two case studies are considered. In all scenarios, the clutter model is assumed to be spatially Poisson distributed with known parameter λ (the number of false measurements per unit of volume $\left(\mathrm{km}^2\right)$) [12, 22]. The target's motion and measurement models are defined by (1) and (2), where state transition matrices F and G, and measurement matrix H are given by [12, 22]:

$$\mathrm{F} = \begin{pmatrix} 1 & \delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{29}$$

$$\mathrm{G} = \begin{pmatrix} \delta/2 & 1 & 0 & 0 \\ 0 & 0 & \delta/2 & 1 \end{pmatrix}^{\mathrm{T}} \tag{30}$$

$$\mathrm{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{31}$$

where δ is the sampling interval, and by using Cartesian coordinates, state vector x containing the position and velocity in x and y is given by:

mmeybodi@aut.ac.ir

$$X(k) = \begin{pmatrix} x(k) \\ v_x(k) \\ y(k) \\ v_y(k) \end{pmatrix} \tag{32}$$

The covariance matrices $Q_{2\times2}$ and $R_{2\times2}$ are respectively the system noise and measurement noise, which are assumed to be $Q_{ii} = (0.02^2)\text{km}^2$ and $R_{ii} = (0.0225)\text{km}^2$ $\left(R_{ij} = Q_{ij} = 0, \text{for } i \neq j\right)$.

To illustrate the performance of FD-JPDAF, the results are compared with JPDAF, MEF-JPDAF [22] and Fuzzy-GA [2]. In simulations of MEF-JPDAF and Fuzzy-GA, the gate probability P_G of these algorithms was set to 0.99 and the detection probability of the true measurement P_D was set to 0.95. To compare the performance of all filters, 100 Monte Carlo runs were performed. The performance of FD-JPDA is compared in terms of RMSE of position and velocity as depicted Table 1.

With FD-JPDAF, 2 parameters need to be set in step 2, i.e. *Eps* and *MinPts*, while parameter ε in step 3 was set to 0.51 [22]. *Eps* and *MinPts* are essential parameters in the DBSCAN algorithm, the exact tuning of which can enhance algorithm performance. Several studies in the past decade have addressed adjusting these parameters [33, 34] for use in FD-JPDAF. However, starting with a prediction point leads to the reduced importance of these parameters.

As mentioned above, *MinPts* is the minimum number of points in a cluster and is set to 3. In fact, any measurement data point with at least 2 neighbours in the vicinity of the target prediction position (or previous core point) is considered as a (new) core point. Many preliminary experiments with various *Eps* have been performed to obtain the optimal value. We have found that, $0.45C \leq Eps \leq 0.6C$ is most effect on the performances of the FD-JPDAF. Where C is the volume of m-dimensional hypersphere validation gate units (in comparing methods) and is set to 0.55C.

4.1 Case 1: Linear Parallel Targets

This case study considered two parallel targets with initial state vectors $x_1(0) = [2550\,\text{m}\ 0.05\,\text{km/s}\ 260\,\text{m}\ 0.05\,\text{km/s}]^T$ and $x_2(0) = [3050\,\text{m}\ 0.05\,\text{km/s}\ 260\,\text{m}\ 0.05\,\text{km/s}]^T$ [2]. The actual and estimated targets trajectories are depicted in Fig. 2. According to Table 1, average performance of FD-JPDAF is improved in comparison with the other algorithms. In fact, the average position RMSE for target-1 is improved by 32%, 5% and 1.5% compared to JPDAF, MEF-JPDAF and Fuzzy-GA, respectively. Whereas, the average position RMSE for target-2 is 36%, 13% and 5.6% compared to JPDAF, MEF-JPDAF and Fuzzy-GA, respectively. Also, FD-JPDAF produced less the average velocity RMSE than the other algorithms and the average velocity RMSE is improved compared to JPDAF and MEF-JPDAF. FD-JPDAF have average velocity RMSE close to Fuzzy-GA.

mmeybodi@aut.ac.ir

4.2 Case 2: Linear Crossing Targets

In second scenario, an example of three crossing targets moving in straight lines is considered [12]. The initial state vectors of the targets are given by $x_1(0) = [1\,\text{km}\; 0.25\,\text{km/s}\; 9.3\,\text{km}\; -0.1\,\text{km/s}]^T$, $x_2(0) = [1\,\text{km}\; 0.25\,\text{km/s}\; 4.3\,\text{km}\; 0.1\,\text{km/s}]^T$ and $x_3(0) = [1\,\text{km}\; 0.25\,\text{km/s}\; 11.3\,\text{km}\; -0.1\,\text{km/s}]^T$. The actual target tracks and the tracks estimated by FD-JPDAF are portrayed in Fig. 3.

According to Table 1, the average position RMSE is improved by 34%, 40% and 2.8% for target-1 and 33%, 35% and 6% for target-2 compared to JPDAF, MEF-JPDAF and Fuzzy-GA, respectively. Whereas the average position RMSE for target-3 is 16% and 26% compared to JPDAF and MEF-JPDAF, respectively. However, FD-JPDAF average position RMSE for target-3 is 2.5% less compared to Fuzzy-GA. Similar to the previous scenario, FD-JPDAF average velocity RMSE is less than the others algorithms.

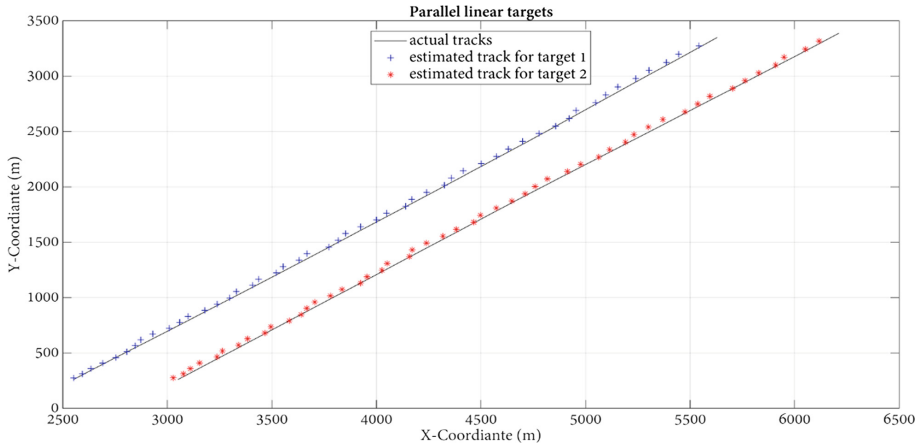


Fig. 2. Actual tracks and tracks estimated by FD-JPDAF for case 1.

As seen in Table 1, increasing clutter density caused a decrease in algorithm performance. Moreover, the most effective increases in clutter density was found for JPDAF, followed by MEF-JPDAF. However, FD-JPDAF and Fuzzy-GA, have similar effect under increase of clutter density. By comparing results, it is evident that the proposed data association's efficiency is comparable to all other existing methods.

mmeybodi@aut.ac.ir

Table 1. Performance comparison in the presence of clutter and false alarms.

Clutter density (λ)	Performance measure	Case 1: linear parallel targets		Case 2: linear crossing targets		
		Target 1	Target 2	Targe 1	Targe 2	Targe 3
JPDAF						
1	Pos.RMSE (m/s)	7.83	7.90	26.43	25.68	37.41
	Vel.RMSE (m/s^2)	1.04	0.99	4.38	4.06	6.82
2	Pos.RMSE (m/s)	8.27	8.42	28.18	26.93	39.03
	Vel.RMSE (m/s^2)	1.31	1.23	4.72	4.52	7.38
MEF-JPDAF						
1	Pos.RMSE (m/s)	5.55	5.82	28.67	26.24	42.69
	Vel.RMSE (m/s^2)	0.84	0.85	4.62	4.58	6.75
2	Pos.RMSE (m/s)	5.93	6.29	30.15	27.86	25.94
	Vel.RMSE (m/s^2)	0.88	0.91	4.94	4.71	7.44
Fuzzy-GA						
1	Pos.RMSE (m/s)	5.35	5.38	17.81	18.28	30.73
	Vel.RMSE (m/s^2)	0.75	0.73	2.89	3.04	5.97
2	Pos.RMSE (m/s)	5.69	5.71	18.62	19.75	34.83
	Vel.RMSE (m/s^2)	0.76	0.75	3.18	3.27	6.34
FD-JPDAF						
1	Pos.RMSE (m/s)	5.27	5.08	17.32	17.14	31.52
	Vel.RMSE (m/s^2)	0.74	0.73	2.83	2.96	5.86
2	Pos.RMSE (m/s)	5.63	5.43	17.94	17.73	32.99
	Vel.RMSE (m/s^2)	0.76	0.76	3.13	3.09	6.23

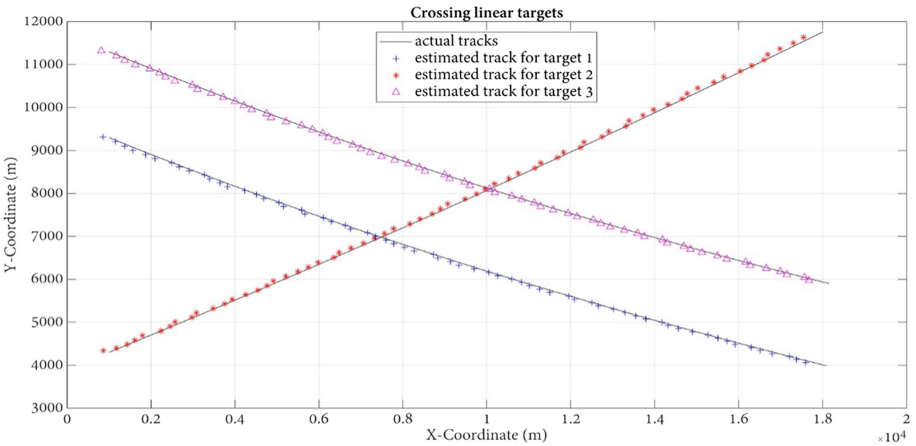


Fig. 3. Actual tracks and tracks estimated by FD-JPDAF for case 2.

mmeybodi@aut.ac.ir

5 Conclusion

In this paper, an efficient and novel data association algorithm named FD-JPDAF was proposed on the basis of density clustering and maximum entropy fuzzy clustering for multi-target tracking. The density clustering approach was used to eliminate noisy measurement and the maximum entropy fuzzy clustering principle was applied to construct an association probability matrix. The effectiveness of the proposed data association approach in multi-target tracking was demonstrated. According to the simulation results, FD-JPDAF outperformed the other filters. Therefore, FD-JPDAF is appropriate for real-time applications and investigating its usage in other applications is a topic for future research.

References

1. Bar-Shalom, Y., Li, X.R.: Multitarget-Multisensor Tracking: Principles and Techniques. YBS Publishing, Storrs (1995)
2. Satapathi, G.S., Srihari, P.: Soft and evolutionary computation based data association approaches for tracking multiple targets in the presence of ECM. Expert Syst. Appl. **77**, 83–104 (2017)
3. Xie, Y., Huang, Y., Song, T.L.: Iterative joint integrated probabilistic data association filter for multiple-detection multiple-target tracking. Digit. Signal Process. **72**, 232–243 (2018)
4. Satapathi, G.S., Srihari, P.: Rough fuzzy joint probabilistic association fortracking multiple targets in the presence of ECM. Expert Syst. Appl. **106**, 132–140 (2018)
5. Bar-Shalom, Y., Fortmann, T.: Tracking, Association, D. & others. Academic Press, San Diego, USA (1988)
6. Collins, J.B., Uhlmann, J.K.: Efficient gating in data association with multivariate gaussian distributed states. IEEE Trans. Aerosp. Electron. Syst. **28**(3), 909–916 (1992)
7. Bergman, N., Doucet, A.: Markov chain Monte Carlo data association for target tracking. In 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings, vol. 2, pp. 705–708. IEEE (2000)
8. Chen, Y.M., Huang, H.C.: Fuzzy logic approach to multisensor data association. Math. Comput. Simul. **52**(5–6), 399–412 (2000)
9. Satapathi, G.S., Srihari, P.: STAP-based approach for target tracking using waveform agile sensing in the presence of ECM. Arab. J. Sci. Eng. **43**(8), 4019–4027 (2018)
10. Aziz, A.M.: A novel all-neighbor fuzzy association approach for multitarget tracking in a cluttered environment. Signal Process. **91**(8), 2001–2015 (2011)
11. Aziz, A.M.: A new nearest-neighbor association approach based on fuzzy clustering. Aerosp. Sci. Technol. **26**(1), 87–97 (2013)
12. Liang-qun, L., Wei-xin, X.: Intuitionistic fuzzy joint probabilistic data association filter and its application to multitarget tracking. Signal Process. **96**, 433–444 (2014)
13. Li, L., Xie, W.: Bearings-only maneuvering target tracking based on fuzzy clustering in a cluttered environment. AEU - Int. J. Electron. Commun. **68**(2), 130–137 (2014)
14. Osman, H.M., Farooq, M., Quach, T.: Fuzzy logic approach to data association. Aerosp./ Defense Sens. Controls **2755**, 313–322 (1996)
15. Singh, R.N.P., Bailey, W.H.: Fuzzy logic applications to multisensor-multitarget correlation. IEEE Trans. Aerosp. Electron. Syst. **33**(3), 752–769 (1997)
16. Smith, J.F.: Fuzzy logic multisensor association algorithm. Proc. SPIE **3068**, 76–88 (1997)

mmeybodi@aut.ac.ir

17. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugen. **7**(2), 179–188 (1936)
18. Nazari, M., Shanbehzadeh, J., Sarrafzadeh, A.: Fuzzy C-means based on automated variable feature weighting. In: Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. I, pp. 25–29, Hong Kong (2013)
19. Chung, Y.N., Chou, P.H., Yang, M.R., Chen, H.T.: Multiple-target tracking with competitive Hopfield neural network based data association. IEEE Trans. Aerosp. Electron. Syst. **43**(3), 1180–1188 (2007)
20. Blackman, S.S., Popoli, R.F.: Design and Analysis of Modern Tracking Systems. Artech House, London (1999)
21. Wang, X., Challa, S., Evans, R.: Gating techniques for maneuvering target tracking in clutter. IEEE Trans. Aerosp. Electron. Syst. **38**(3), 1087–1097 (2002)
22. Liangqun, L., Hongbing, J., Xinbo, G.: Maximum entropy fuzzy clustering with application to real-time target tracking. Signal Process. **86**(11), 3432–3447 (2006)
23. Blackman, S.S.: Multiple-target Tracking with Radar Applications, 463 p. Artech House, Inc., Dedham (1986)
24. Bar-Shalom, Y., Fortmann, T. E. Tracking and Data Association. Academic Press Professional Inc., (1988)
25. Kriegel, H.-P., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **1**(3), 231–240 (2011)
26. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
27. Tran, T.N., Drab, K., Daszykowski, M.: Revised DBSCAN algorithm to cluster data with dense adjacent clusters. Chemom. Intell. Lab. Syst. **120**, 92–96 (2013)
28. Bordogna, G., Ienco, D.: Fuzzy core DBScan clustering algorithm. In: Communications in Computer and Information Science, CCIS, vol. 444, pp. 100–109 (2014)
29. Mahesh Kumar, K., Rama Mohan Reddy, A.: A fast DBSCAN clustering algorithm by accelerating neighbor searching using groups method. Pattern Recognit. **58**, 39–48 (2016)
30. Birant, D., Kut, A.: ST-DBSCAN: an algorithm for clustering spatial–temporal data. Data Knowl. Eng. **60**(1), 208–221 (2007)
31. Liu, P.X., Meng, M.Q.H.: Online data-driven fuzzy clustering with applications to real-time robotic tracking. IEEE Trans. Fuzzy Syst. **12**(4), 516–523 (2004)
32. Zhang, J., Ji, H., Ouyang, C.: Multitarget bearings-only tracking using fuzzy clustering technique and Gaussian particle filter. J. Supercomput. **58**(1), 4–19 (2011)
33. Smiti, A., Elouedi, Z.: DBSCAN-GM: an improved clustering method based on Gaussian Means and DBSCAN techniques. In: 2012 IEEE 16th International Conference on Intelligent Engineering Systems, pp. 573–578, IEEE (2012)
34. Karami, A., Johansson, R.: Choosing DBSCAN parameters automatically using differential evolution. Int. J. Comput. Appl. **91**(7), 1–11 (2014)

mmeybodi@aut.ac.ir

Representation Learning Techniques: An Overview

Hassan Khastavaneh(✉) and Hossein Ebrahimpour-Komleh

University of Kashan, Kashan, Esfahan, Iran
khastavaneh@hotmail.com, ebrahimpour@kashanu.ac.ir

Abstract. Representation learning techniques, as a paradigm shift in feature generation, are considered as an important and inevitable part of state of the art pattern recognition systems. These techniques attempt to extract and abstract key information from raw input data. Representation learning based methods of feature generation are in contrast to handy feature generation methods which are mainly based on the prior knowledge of expert about the task at hand. Moreover, new techniques of representation learning revolutionized modern pattern recognition systems. Representation learning methods are considered in four main approaches: sub-space based, manifold based, shallow architectures, and deep architectures. This study demonstrates deep architectures are considered as one of the most important methods of representation learning as they cover more general priors of real-world intelligence as a necessity for modern intelligent systems. In other words, deep architectures overcome limitations of their shallow counterparts. In this study, the relationships between various representation learning techniques are highlighted and their advantages and disadvantages are discussed.

Keywords: Representation learning · Feature generation · Manifold learning · Shallow architectures · Deep learning

1 Introduction

Feature generation as an essential stage in the pipeline of any typical pattern recognition system is the process of extraction and abstraction of key information from raw sensory data in a way that extracted features represent and describe real-world observations as accurate as possible. Performance of such systems heavily depends on the quality of generated features. If the quality of generated features is adequate, building high-performance regressors and classifiers will be a simple task. Low dimensionality and simplicity are two factors of feature quality; low dimensional features prevent curse of dimensionality and simplicity leads to build simple predictors and consequently more general models. There are two major directions for feature generation: handy feature engineering and representation learning (RL).

Handy feature engineering methods of feature generation usually produce a set of transformed features by applying a transformation with some fixed base functions on the raw data. As the base functions of the transformation are usually chosen by an expert with prior knowledge about the problem at hand, these methods are referred to

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 89–104, 2020.
https://doi.org/10.1007/978-3-030-37309-2_8
mmeybodi@aut.ac.ir

as handy. In addition, handy features have some properties corresponding to the used base functions. The main shortcomings of handy feature engineering methods are high computational cost and inability to extract enough discriminatory information from the raw data. Moreover, for a typical pattern recognition system, selection of handy feature generation methods and setting their corresponding parameters are usually based on trial and error.

Building pattern recognition systems based on handy feature engineering methods cause such systems to depend on the feature generation stage. In order to make pattern recognition systems more robust to the feature generation stage, this dependency should be removed and change it into an automated process. In this context, automation means that the base functions of the transformation should not be fixed but, they should be learned via a training process based on the available data without expert intervention. The solution of automated feature generation or learning features directly from the data is summarized in the RL methods of feature generation. The ultimate goal of RL methods of feature generation is to learn the generation of usable features directly from the raw data in a way that learned features guarantee the best representation. In another perspective, RL methods allow a typical pattern recognition system to be directly fed with the raw sensory data without prior generation of handy features.

RL or feature learning is the task of finding a transformation of raw data in a way to improve the performance of machine learning tasks such as regression and classification. In fact, RL is absolutely essential for approaching real artificial intelligence. Moreover, RL is commonly considered as a potential candidate solution for numerous complex problems of data science. Furthermore, RL methods attempt to make some important concepts of real-world intelligence possible. As mentioned by Bengio and LeCun [1], the most important reason that makes some methods of RL successful is their ability to utilize some general priors related to real-world intelligence. Some of these priors include smoothness, multiple explanatory factors, the sparsity of features, transfer learning, independence of features, natural clustering and distributed representation, semi-supervised learning, and hierarchical organization of features [2]. A typical RL method will be more powerful and valuable if it covers a larger set of the above mentioned general priors.

As there are a variety of RL methods, different categorization of them is manageable. One possibility is to categorize RL methods into four main approaches, including sub-space based RL approaches which look for representations in the subspaces of the original feature space, manifold based RL approaches that represent raw data based on the embedded manifold hidden in the original space, shallow RL approaches, and deep RL approaches.

It is possible to consider RL methods in term of using or not using supervisory information for generating representations. Majority of RL methods such as principal component analysis (PCA), independent component analysis (ICA), restricted Boltzmann machines (RBM) perform unsupervised RL thus, they do not incorporate any class label or other supervisory information in the process of learning representations. In contrast to unsupervised RL methods, supervised RL methods like linear discriminant analysis (LDA) family, incorporate supervisory information in the process of learning representations. However, there are some RL methods that are naturally unsupervised but, they use additional information in the process of learning

mmeybodi@aut.ac.ir

representations; hence, they are called soft supervised RL methods. Semi-supervised RL methods utilize both labeled and unlabeled data for generating representations. Worth to mention that the main focus of RL methods is on the unsupervised and semi-supervised methods of feature generation.

It is supposed that RL is the task of looking for a transformation (mapping) function $f: X^D \rightarrow Y^d$, which transforms (maps) data from the original feature space X, with the dimension D, to the representation space Y, with the dimension d. Dimensionality of the representation space is usually much smaller than the dimensionality of original feature space. As an exception, in order to force generated representations to have a specific property, their dimension may be much greater than the dimension of data in the original space. Moreover, in some methods of RL like convolutional neural networks for classification, the output (results of mapping) is an encoding which is consistent with the final output of the pattern recognition system. In other words, the final output of the transformation is the predicted value of such tasks. Some RL methods have intermediate transformations and consequently representations organized into multiple layers. Such representation methods with multiple hierarchical layers are elaborated in the later sections.

Rest of this paper is organized as follows: sub-space based RL approaches are explained in Sect. 2, Sect. 3 discusses manifold based RL approaches, shallow RL approaches are discussed in Sect. 4, Sect. 5 explains deep RL approaches, Sect. 6 discus RL approaches and conclude the paper.

2 Sub-space Based Representation Learning Approaches

Sub-space based approaches as almost early methods of RL attempt to look for a sub-space in the original feature space that better represent the original data. This representation is achieved by projecting data of the original feature space into new sub-space by applying the learned transformation function; the generated representation has some properties corresponding to the way base functions of the transformation are formed. In sub-space based RL methods, new features are commonly generated by a linear combination of original features thorough base functions; the base functions of transformation are learned by analyzing data in the original feature space. During the learning process of the base functions, independence, orthogonality, and sparsity as potential properties may be obtained. In the sections ahead, the most popular sub-space based RL methods, including PCA family, metric multi-dimensional scaling (MDS), ICA family, and LDA family, are considered.

2.1 Principal Component Analysis Family

PCA as a global method is one of the oldest techniques of unsupervised data representation which focus on the orthogonality of generated features [3]. The main purpose of PCA is to generate a low dimensional representation of the original observations and preserve maximum variance of the original data as well. The base functions of transformation are actually principal components hidden in the original data. A solution for finding transformation matrix is to use a portion of eigenvectors of the covariance

mmeybodi@aut.ac.ir

matrix of the original data. The number of selected eigenvectors determines the dimension of the new representation. The eigenvalue corresponding to each eigenvector measures its importance in term of the amount of held variance.

PCA is suffering from the fact that the principal components are created by an explicit linear combination of all of the original observations. This phenomenon does not allow to interpret each principal component independently. In order not to use all of the original variables is to utilize Sparse PCA (SPCA) which reduce the dimensionality of the data by adding sparsity constraint to the original variables [4].

As it is the case in many real-world applications, if the generation mechanism of data is non-linear, the original PCA fails to recover true intrinsic dimensionality of the data. This is considered a shortcoming of PCA which is relieved by its kernelized version known as Kernel PCA (KPCA) [5].

It is also possible to derive PCA within a density estimation framework based on a probability density model of the observed data. In this case, the Gaussian latent-variable model is utilized to derive probabilistic formulation of PCA. Latent-variable formulation of obtaining principal axes leads naturally to an iterative and computationally efficient expectation-maximization solution for applying PCA commonly known as Probabilistic PCA [6].

2.2 Metric Multidimensional Scaling

Metric multidimensional scaling is a linear technique for generating representations. In contrast to PCA which project data into a sub-space that preserves maximum variance, MDS project data into a sub-space which preserve pairwise squared distance. In other words, MDS attempts to preserve the dot product of samples in the new representation space [7]. The idea of distance preservation used in MDS has been used in one way or another in some manifold learning. As Eigen decomposition of the Gram matrix which holds pairwise dot product of samples is required for MDS, kernel PCA can be considered as a kernelized version of MDS, where the inner product in the input space is replaced by kernel operation in the Gram matrix.

2.3 Independent Component Analysis Family

ICA is another popular technique of sub-space based RL which is very similar to PCA. In contrast to PCA which uses variance as second-order statistical information, ICA uses higher order statistics for generating representations. Using higher order statistics force generated features to be mutually independent [8]. In a topological variation of ICA, independence assumption of generated features is removed and a degree of dependence based on the distance between generated features is assigned. Mentioned distances lead to generate a topological map which is used by some applications of computer vision [9]. Kernel ICA is another variation of original ICA which uses calculated correlation in the reproducible kernel Hilbert space for generating non-linear representations [10].

mmeybodi@aut.ac.ir

2.4 Linear Discriminant Analysis Family

LDA is a global and supervised method of RL. In this method, the transformation matrix is obtained in a way to generate features that hold maximum variance an also bring maximum class separability by utilizing the within-class and the between-class amount of variances exist in the data. In other words, the transformation matrix is computed in a way that the amount of between-class variance relative to the amount of within class variance is maximized. Generating features that satisfy class separability property is desirable for many applications [11]. An incremental version of LDA is also proposed for those applications which demand generated representation space be updated at the arrival of new data sample [12].

To conclude sub-space based RL methods, many methods try to find sub-space in one way or another. This sub-space has some properties that are transferred to the generated features. The advantage of sub-space methods of representation generation is computational efficiency thanks to eigen decomposition technique. As sub-space methods are linear in nature, they cannot be successful when the original data are generated non-linearly. In the case of non-linearity, for better representation, other RL methods such as manifold family are potential candidates to be considered in the next section.

3 Manifold Based Representation Learning Approaches

Among the family of RL approaches, manifold based methods have attracted attention due to their nonlinear nature, geometrical intuition, and computational feasibility. A strong assumption in most manifold learning methods is that the data appears in the original high dimensional feature space approximately belongs to a manifold with an intrinsic dimension less than the dimension of original space. In other words, the manifold is embedded in the original high dimensional feature space. The goal of manifold based RL methods is to find this low dimensional embedding and consequently generating a new representation of original observations based on the founded embedding. In contrast to sub-space based RL approaches which usually perform dimensionality reduction and consequently linear RL, manifold based approaches reduce the dimension in a nonlinear fashion by attempting to uncover intrinsic low-dimensional geometric structures hidden in the original high dimensional observation space.

Manifold based RL methods are categorized into three main groups of local, global, and hybrid; each method attempts to preserve different geometrical properties of the underlying manifold while attempting to reduce the dimension of original data.

3.1 Local Methods of Manifold Learning

Local manifold learning methods attempt to capture local interactions of samples in the original feature space and transfer captured interactions to the generated new low-dimensional representation space. The strategies followed by local methods of manifold learning lead to map nearby points of the original feature space to nearby points in

mmeybodi@aut.ac.ir

the newly generated low-dimensional representation space. Computational efficiency and representation capacity are two characteristics of local methods. Computations of local methods are efficient because the matrix operands that exists in local methods are usually sparse.

Laplacian eigenmaps [13], local linear embedding (LLE) [14], and Hessian eigenmaps [15] are representative methods of local manifold learning family. Laplacian eigenmaps captures local interactions of data by utilizing Laplacian of the original data graph. Sensitivity to noise and outliers are considered as a shortcoming of Laplacian eigenmaps. Representations generated by LLE are invariant under rotation, translation, and scaling as geometrical transformations. Hessian eigenmaps is the only method of manifold learning capable of dealing with non-convex data. As all the methods based on Hessian operator needs to calculate second derivatives, they are sensitive to noises, especially in high dimensional data.

3.2 Global Methods of Manifold Learning

The fact that representations generated by global methods of manifold learning cause the nearby points to remain nearby and also faraway points remain faraway, tends these methods to give more faithful representation than local methods.

Isometric feature mapping or shortly ISOMAP is the most popular global method of manifold learning. ISOMAP uses the geodesic distance between all pairs of the data points to uncover the true structure of the manifold. Using geodesic distance instead of Euclidean distance leads faraway points in the original space to remains faraway in the representation space. The reason for this desirable property is that some points that are close in term of Euclidean distance may be far in term of geodesic distance. In fact, geodesic distance allows learning the global structure of the data. ISOMAP is also considered as a variant of the MDS algorithm in which the Euclidean distances are changed to the Geodesic distances along the manifold [16].

Experimental result demonstrates ISOMAP cannot scale well for large datasets as it demands huge amounts of memory for storing distance matrices. In order to increase its scalability, landmark ISOMAP (L-ISOMAP) has been proposed by using a subset of data points known as landmark points [17].

3.3 Hybrid Methods of Manifold Learning

As mentioned previously, both local and global methods of manifold learning have their own advantages and disadvantages in terms of representation capability and computational efficiency. Hybrid methods of manifold learning usually attempt to globally align local manifolds and gain benefits of computational efficiency of local methods and quality representation generation of global methods. In other words, hybrid methods generate representations approximately as good as global methods by an efficient cost of local methods. Some of the well-known hybrid methods of manifold learning are conformal ISOMAP [17], manifold charting [18], and diffusion maps [19].

To conclude, manifold based methods of RL exist in different categories with different properties. Early local methods are sensitive to noises and outliers. Moreover, proper parameter tuning is mandatory for some methods. Experiments demonstrate

mmeybodi@aut.ac.ir

global methods of manifold learning gives a better representation than local methods. However, this excellence comes with a higher cost of computation. As the computational cost of local methods is more reasonable, some hybrid methods attempt to follow the path of local methods for obtaining representations with the capability as close as global methods. Some manifold learning methods have a close relationship to sub-space based methods such as MDS and Kernel PCA. Despite many progress in manifold learning methods, the problem of manifold learning from noiseless and sufficiently dense data still remains a difficult challenge. Although manifold learning methods generate representations better than sub-space based approaches, still we need better methods for generating representations that meet the requirements of real-world intelligence.

4 Shallow Representation Learning Approaches

The focus of this section is the consideration of shallow RL approaches in term of representation capability and computational efficiency. As a matter of fact, sub-space and manifold based RL approaches are under the umbrella of shallow architectures. Also, some machine learning techniques such as multilayer perceptron with less than five layers and local kernel machines are considered as shallow architecture methods; these techniques generate a limited representation of input data in their mechanism prior to producing any prediction output.

In order to represent any function or learn behavior and underlying structure of any data by using shallow architectures, an exponential number of computational elements with respect to the input dimension is required. As a result, shallow methods are not compact enough. Compactness means fewer computational elements and consequently fewer free parameter tuning. Accordingly, non-compact nature of shallow methods of RL lead these methods to have poor generalization property.

As the majority of shallow architecture RL methods are indeed local estimators, they exhibit poor generalization while learning highly varying functions. The reason for lack of generalization, in this case, is that local estimators partition input space into regions whose number relates to the number of variations in the target function. Each partition needs its own parameters for learning the shape of that region. As a result, much more training examples are needed to support the training of variations in the target function. Kernel machines and many unsupervised RL methods such as ISO-MAP, LLE, and Kernel PCA are good examples of local estimators which are considered as shallow architecture RL techniques. In order to tackle limitations of kernel machines as local estimators, some techniques are needed to learn better feature space and consequently learning highly varying target functions in an efficient manner. Worth to mention, if the variations of target function are independent, no learning algorithm will perform better than local estimators [20]. Restricted Boltzmann machines (RBM) and autoencoders as shallow architecture methods of RL are introduced in the sections ahead.

mmeybodi@aut.ac.ir

4.1 Restricted Boltzmann Machines

Restricted Boltzmann machines (RBMs) are actually energy-based probabilistic graphical models which attempt to learn the distribution of input data. As Fig. 1 depicts, a typical RBM has two layers of visible and hidden nodes. The visible layer nodes are connected to the hidden layer nodes via weight matrix W. There are no visible-visible and hidden-hidden connections hence, these types of Boltzmann machines are so-called restricted. RBMs are able to compactly represent any distribution in case of providing enough hidden nodes. The scaler energy associated to each configuration of the nodes in a typical RBM is defined by Eq. 1 as energy function and the probability distribution via mentioned energy function is described by Eqs. 2, 3, and 4. Here, b and c refer to the biases of visible and hidden nodes respectively [21].

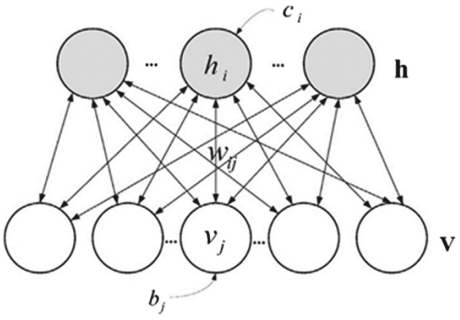


Fig. 1. The architecture of a typical restricted Boltzmann machine [22].

$$E(v.h) = -\acute{b}v - \acute{c}h - \acute{h}Wv \tag{1}$$

$$p(x) = \frac{e^{-\mathcal{F}(x)}}{Z} \tag{2}$$

$$Z = \sum\nolimits_x e^{-\mathcal{F}(x)} \tag{3}$$

$$\mathcal{F}(x) = -\log \sum\nolimits_h e^{-E(x.h)} \tag{4}$$

In order to learn the desired configuration, the energy function should be modified through a stochastic gradient descend procedure on the empirical negative log-likelihood of the data-set whose distribution needs to be learned. Equations 5 and 6 defines required log-likelihood and loss functions respectively. In these equations, θ and $\mathcal{D}$ refers to the model parameters and training data respectively. The parameter set (θ) which needs to be optimized include, weight matrix W, biases of visible nodes b, and biases of hidden nodes c. Gradient of negative log likelihood as described by Eq. 7 has two terms refereed as positive and negative phases. Positive phase deals with the

mmeybodi@aut.ac.ir

probability of the training data while, negative phase deals with probability of samples generated by the model itself. The negative phase allows to check what have been learned by the model up to current iteration. In order to make computation of the gradient tractable, the expectation of all possible configuration of visible nodes v under model distribution P is estimated via a fixed number of model samples known as negative particles. The negative particles $\mathcal{N}$ are sampled from P by running a Markov chain with Gibbs sampling as its transition operator. In order to efficiently optimize model parameters, contrastive divergence (CD) is utilized. CD-*k* initialize the Markov chain using one of the training examples and limits the transition just to *k* step. Experimental results demonstrate the value *1* for *k* is appropriate for learning data distribution [23]. For better performance, construction and training of RBMs need some proper settings, including the number of hidden units, the learning rate, the momentum, the initial values of weights, the weight-cost, and the size of mini batches of the gradient descent. To clarify the effect of these meta-parameters on each other, by having more hidden nodes, the representation capacity of RBMs increases with the cost of increasing training time. In addition, types of units to be used and decision on whether to update the states of each node stochastically or deterministically are important [24].

$$\mathcal{L}(\theta.\mathcal{D}) = \frac{1}{N}\sum\nolimits_{x^{(i)}\in\mathcal{D}} \log p\left(x^{(i)}\right) \tag{5}$$

$$\ell(\theta.\mathcal{D}) = -\mathcal{L}(\theta.\mathcal{D}) \tag{6}$$

$$-\frac{\partial \log p(x)}{\partial\theta} \approx \frac{\partial\mathcal{F}(x)}{\partial\theta} - \frac{1}{|\mathcal{N}|}\sum\nolimits_{\tilde{x}\in\mathcal{N}} \frac{\partial\mathcal{F}(\tilde{x})}{\partial\theta} \tag{7}$$

As the training of a typical RBM is converged, it is ready to generate a new representation in the hidden layer for any data presented to its visible layer. RBMs are also considered as multi-clustering methods which are a kind of distributed representation. Distributed representation as a requirement for real-world intelligence is the capability which leads each hidden node concerns one specific aspect of the data which have been presented to its visible nodes. Distributed representation of RBMs enable generalization to a new combination of values of learned features beyond those have been seen during its training. RBMs are used in a variety of applications including analysis of complex computer tomography images [25].

4.2 Autoencoders

Autoencoders are actually unsupervised neural networks trained via back-propagation algorithm with the setting that target values are the input values [26]. A typical autoencoder is composed of an encoding unit that generates representations, decoding unit that reconstructs input from representation, and one hidden or representation layer which desired to captures main factors of variations hidden in the data. Early autoencoders attempt to learn a function which is an approximation to the identity function.

mmeybodi@aut.ac.ir

By applying some constraints on the autoencoder network and specifically its objective function, more interesting structures hidden in the data will be discovered. These constraints usually appear in different forms of regularization. Simplest regularization technique is the weight decay which forces the weights to be as small as possible. Going from linear hidden layer to nonlinear one leads the autoencoders to capture multi-modal aspects of the input distribution [27]. Sparsity is a solution for preventing autoencoders from learning the identity function. In this setting, which is known as over-complete setting, the size of hidden layer is greater than the size of the input layer and many of the hidden nodes get zero or near zero values [28]. In order to force the hidden layer to learn more robust and generalized representation, denoising autoencoders that lead the network to learn representation from a corrupted or noisy version of the data are proposed. Representations generated from noisy data are more robust than their previous counterparts [29]. Variational autoencoders (VAEs) as a generative variation of autoencoder networks, attempt to generate new samples to exploring variations hidden in the data. In contrast to other methods of sample generation which are random, VAEs generate samples in the direction of existing data to fill the gaps in the latent space thanks to their continuous latent space [30].

5 Deep Representation Learning Approaches

Deep architectures are among potential solutions for tackling previously mentioned limitations of shallow RL approaches. As deep architectures of RL cover more general priors of real-world intelligence, they are considered as the most promising paradigms for solving complex real-world problems of artificial intelligence up to know. In other words, multiple layers of representation in deep architectures facilitate the reorganization of feature space that causes machine learning methods to learn highly varying target functions. Deep RL methods are necessary for AI-level applications which need to learn complicated functions that represent high-level abstractions. Deep representations are obtained by utilizing deep architectures that are the composition of multiple stacked layers. These multiple processing layers attempt to automatically discover abstractions from lowest level observations to the highest level concepts. Abstractions in different layers allow building concept hierarchy as a necessity for real-world intelligence. In other words, higher layers attempt to amplify important aspects of raw data and suppress irrelevant variations [31].

Neural networks are considered as the most promising path for approaching deep RL. A typical deep neural network (DNN) is actually a network with multiple stacking layers of simple non-linear processing units. Because of the large number of layers and units per layer, training of such large networks demands a huge number of training data and computational power for better generalization.

Training of a typical DNN is commonly based on error gradient back propagation which relies on multiple passes over training data. As the number of parameters in DNNs is huge, too many training data and consequently long iterations are needed for proper optimization. In order to decrease the training time of DNNs as a large scale machine learning problem, stochastic gradient descend (SGD) has been proposed [32].

mmeybodi@aut.ac.ir

Training of deep neural networks is a difficult optimization problem because of vast parameter space with too many local optima and plateau which their computed gradient is zero. In order to train DNNs, layer-vise unsupervised pre-training, convolution, auto-associators, dropout, and other techniques are utilized. These techniques cause construction of special types of deep neural networks namely, deep belief networks, convolutional neural networks, deep auto-encoding networks, and dropout networks respectively.

5.1 Deep Belief Networks

One solution for preventing neural networks from getting stuck in the points of parameter space with zero gradients is to initialize the weights in an unsupervised manner prior to fine-tuning of the network weights [33]. In this strategy, the network is built by stacking multiple layers of feature detector which are individually trained using unlabeled data. After stacking multiple pre-trained layers, the whole network is fine-tuned using the standard back-propagation algorithm. Such networks also prevent over-fitting in cases of the small labeled dataset which allows having a more generalized model [34].

Deep belief networks (DBNs) are one of those network types which beneficiary from unsupervised pre-training. A belief network is actually a directed acyclic graph composed of stochastic variables. In fact, the layers that build deep belief networks are RBMs; thus, a typical DBN is actually a layer-wise composition of multiple probability density models instead of mixture or product of those models.

5.2 Convolutional Neural Networks

Another solution to remedy problems related to the training of deep neural networks is the concept of convolution which allows sharing network weights. The shared weights allow having much smaller network free parameters than fully connected networks and consequently better network training and generalization. As an advantage, the small number of free parameters allow having deeper networks [35]. In addition, weight sharing via convolution concept allows having networks with the capability of translation invariance. Such networks which are inspired by biological processes of vision in animals are called convolutional neural networks (CNNs). A typical CNN in addition to the input and output has multiple hidden layers of convolution, pooling, and fully-connected layers as essential building blocks. In addition to the mentioned layer types, some networks may have special types of layers for performance improvement on a specified task.

Convolution layers in a CNN allow applying n-dimensional filters commonly known as kernels on multi-dimensional unstructured data. Multi-dimensional filters let to exploit topological information exist in different channels of visual data. In addition, the organization of these features using multiple layers allow having a hierarchy of features from the raw to the more abstract and meaningful ones.

Pooling layers as another important building block of CNNs attempt to reduce the size of feature maps. Pooling mechanism is very similar to the convolution mechanism but, instead of linear combination performed on sub-regions, the values are passed

mmeybodi@aut.ac.ir

through a pooling function. Pooling as a mechanism of down-sampling provides a form of translation invariance. The semantic merging of similar features into one via pooling allows having more compact and robust representation.

Fully-connected layers in a CNN which are usually placed near to the output layer, connect all of the output layer nodes to all of the nodes in its previous adjacent layer. Indeed, these layers carry the task of high-level reasoning.

Increasing the depth of network cause to increase network accuracy but, if the number of layers exceeds a certain value, accuracy becomes saturated in the training phase. In order to address degradation in accuracy of training, residual networks have been proposed. In such networks, the layers learn residual functions $\mathcal{F}(x) = \mathcal{H}(x) - x$ with reference to the layer input. Here, $\mathcal{H}(x)$ is a mapping from a set of layers and x is the input of the first layer in the set. Experiments demonstrate the definition of residual functions on a single layer has no advantage thus, residual functions are usually explicitly defined on a set of layers [36].

The most important real-word priors that CNNs cover are the hierarchy of representation and transfer learning. It is possible to train a CNN on the source task with many images and transfer the network with the learned features to the target task with fewer training data. CNNs have wide applications, including face recognition [37], human action recognition, diagnosis of Helicobacter pylori infection [38], brain segmentation [39], diagnosis of breast cancer [40], lung nodule classification [41], object detection [42], and image recognition [36].

5.3 Dropout Networks

Dropout technique attempts to prevent deep neural networks from over-fitting by randomly dropping some units from the network in the training phase. This droption prevent the units from being too much adopted and consequently over-fitted. As a matter of fact, dropout allows sampling of a large number of diverse sub-networks during the training. As the number of free parameters of these sub-networks is less in comparison with the original network, they have less tendency to be over-fitted. In another perspective, because the architecture of these sub-networks is different, they behave like ensemble methods [43] and consequently as it is expected, the performance improvement is achieved. As the parameter updates are very noisy, training time is very slow which is considered as a drawback of dropout networks. Dropout technique is widely used in various types of deep networks such as CNNs and DBNs. Dropout is used for fine-tuning of DBNs as well.

5.4 Deep Autoencoders

Deep autoencoders actually are an extension of early shallow autoencoders. By adding more layers to the encoding and decoding units of shallow autoencoders, their representation capability is improved. These layers in combination with convolution technique enable autoencoders to handle unstructured data such as images. The deep version of many variations of autoencoders such as denoising autoencoders also has been proposed [44]. Deep autoencoders have a variety of applications, including image compression [45] and content-based image retrieval [46].

mmeybodi@aut.ac.ir

6 Discussion and Conclusion

Generating features via RL techniques is more useful than handy feature generation; this study reveals many efforts have been done from the past to present for proposing better techniques of representation generation. These techniques range from early and simple sub-space based methods to the more sophisticated methods of deep architectures. In fact, advanced methods of feature generation are mandatory for any intelligent system as they offer useful representations from the raw data.

As justified previously, sub-space methods of representation generation are computationally efficient thanks to Eigen-decomposition techniques; but, they cannot perform well in situations that the data are generated in a non-linear fashion. In contrast to sub-space based methods, manifold-based approaches are capable to generate representations in cases of non-linear data. Alongside the capability for handling non-linear data, sensitivity to noise and outlier is a problem that some manifold learning methods such as Laplacian eigenmaps and Hessian eigenmaps are suffering from. Moreover, despite many signs of progress in the development of manifold learning methods, the problem of manifold learning from noiseless and sufficiently dense data still remains a difficult challenge. In addition, both sub-space methods and manifold-based methods are categorized as shallow architectures with limited representation capabilities.

As there are various methods of RL with their own advantages and disadvantages, the methods based on the deep architectures are considered as the most complete ones; the reason for this completeness is the fact that they cover more general priors of real-world intelligence [1]. One of the most important prior which deep architectures cover is the hierarchical organization of features which allows building high-level features on the top of low-level features by multiple abstractions in different layers. Moreover, passing data through a system with multiple layers allow to strength relevant features and suppress irrelevant ones. Moreover, transfer learning is also another prior which brings artificial intelligent agents close to the real world intelligent agents; deep architectures are capable to learn the concepts from the data of source task via their multiple layers and transfer those concepts to the target task.

Convolutional neural networks as the most successful techniques of deep architectures allow to abstract and extract features from raw unstructured data. Autoencoders can use convolution layers for their encoding and decoding parts. As autoencoders perform manifold learning, convolutional autoencoders are very useful for feature generation without using supervisory information. Research in the area of deep RL methods is continuing to offer new network architectures with the highest performance for different applications.

By emerging RL, much effort is on the improvement of feature space instead of classification techniques. In other words, there are very successful classification techniques that need better organization of input features for real-world level intelligent applications.

mmeybodi@aut.ac.ir

References

1. Bengio, Y., Lecun, Y.: Scaling learning algorithms towards AI. In: Large Scale Kernel Machines, pp. 321–360 (2007)
2. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 1798–1828 (2012)
3. Cadima, J., Jolliffe, I.T.: Loading and correlations in the interpretation of principle components. J. Appl. Stat. **22**, 203–2014 (1995)
4. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. J. Comput. Graph. Stat. **15**, 262–286 (2006)
5. Schölkopf, B., Smola, A., Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**, 1299–1319 (1998)
6. Zhao, J., Philip, L.H., Kwok, J.T.: Bilinear probabilistic principal component analysis. IEEE Trans. Neural Netw. Learn. Syst. **23**, 492–503 (2012)
7. Abdi, H.: Multidimensional scaling: eigen-analysis of a distance matrix. In: Encyclopedia of Measurement and Statistics, pp. 598–605 (2007)
8. Comon, P.: Independent component analysis, a new concept? Sig. Process. **36**, 287–314 (1994)
9. Hyvärinen, A., Hoyer, P.O., Inki, M.: Topographic independent component analysis. Neural Comput. **13**, 1527–1558 (2001)
10. Bach, F.R., Jordan, M.I.: Kernel independent component analysis. J. Mach. Learn. Res. **1**, 1–48 (2002)
11. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Ann. Eugen. **7**, 179–188 (1936)
12. Aliyari Ghassabeh, Y., Rudzicz, F., Moghaddam, H.A.: Fast incremental LDA feature extraction. Pattern Recognit. **48**, 1999–2012 (2015)
13. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**, 1373–1396 (2003)
14. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
15. Donoho, D.L., Grimes, C.: Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. In: Proceedings of the National Academy of Sciences, pp. 5591–5596 (2003)
16. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
17. De Silva, V., Tenenbaum, J.B.: Global versus local methods in nonlinear dimensionality reduction. In: Proceedings of the 15th International Conference on Neural Information Processing Systems, pp. 721–728. MIT Press, Cambridge (2002
18. Brand, M.: Charting a manifold. In: Advances in Neural Information Processing Systems, pp. 961–968 (2002)
19. Coifman, R.R., Lafon, S.: Diffusion maps. Appl. Comput. Harmon. Anal. **21**, 5–30 (2006)
20. Bengio, Y.: Learning deep architectures for AI. Found. Trends® Mach. Learn. **2**, 1–127 (2009)
21. Freund, Y., Haussler, D.: Unsupervised learning of distributions on binary vectors using two layer networks. In: Advances in Neural Information Processing Systems, pp. 912–919 (1992)
22. Zhang, C.-Y., Chen, C.L.P., Chen, D., Ng, K.T.: MapReduce based distributed learning algorithm for restricted Boltzmann machine. Neurocomputing **198**, 4–11 (2016)

mmeybodi@aut.ac.ir

23. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Comput. **1800**, 1771–1800 (2002)
24. Hinton, G.E.: A practical guide to training restricted Boltzmann machines. Neural Net.: Tricks Trade **7700**, 599–619 (2012)
25. Van Tulder, G., De Bruijne, M.: Combining generative and discriminative representation learning for lung CT analysis with convolutional restricted Boltzmann machines. IEEE Trans. Med. Imaging **35**, 1262–1272 (2016)
26. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
27. Japkowicz, N., Hanson, S.J., Gluck, M.A.: Nonlinear autoassociation is not equivalent to PCA. Neural Comput. **12**, 531–545 (2000)
28. Ranzato, M.A., Poultney, C., Chopra, S., Cun, Y.L.: Efficient learning of sparse representations with an energy-based model. In: Advances in Neural Information Processing Systems, pp. 1137–1144. MIT Press (2007)
29. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning – ICML, pp. 1096–1103. ACM Press, New York (2008)
30. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations (ICLR), pp. 1–14 (2014)
31. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
32. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT, pp. 177–186 (2010)
33. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems, pp. 153–160. MIT Press (2007)
34. Erhan, D., Courville, A., Vincent, P.: Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. **11**, 625–660 (2010)
35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)
36. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE (2016)
37. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Computer Vision and Pattern Recognition (CVPR), pp. 815–823. IEEE (2015)
38. Shichijo, S., Nomura, S., Aoyama, K., Nishikawa, Y., Miura, M., Shinagawa, T., Takiyama, H., Tanimoto, T., Ishihara, S., Matsuo, K., Tada, T.: Application of convolutional neural networks in the diagnosis of helicobacter pylori infection based on endoscopic images. EBioMedicine **25**, 106–111 (2017)
39. Chen, H., Dou, Q., Yu, L., Qin, J., Heng, P.-A.: VoxResNet: deep voxelwise residual networks for brain segmentation from 3D MR images. Neuroimage **170**, 446–455 (2018)
40. Motlagh, M.H., Jannesari, M., Aboulkheyr, H., Khosravi, P.: Breast cancer histopathological image classification: a deep learning approach, pp. 1–8 (2018)
41. Yuan, Y., Chao, M., Lo, Y.-C.: Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance. IEEE Trans. Med. Imaging **36**, 1876–1886 (2017)
42. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525. IEEE (2017)
43. Rokach, L.: Ensemble-based classifiers. Artif. Intell. Rev. **33**, 1–39 (2010)

mmeybodi@aut.ac.ir

44. Vincent, P., Larochelle, H.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion pierre-antoine manzagol. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
45. Salakhutdinov, R., Hinton, G.: Semantic hashing. Int. J. Approx. Reason. **50**, 969–978 (2009)
46. Krizhevsky, A., Hinton, G.: Using very deep autoencoders for content-based image retrieval. In: Proceedings of the European Symposium on Artificial Neural Networks, pp. 1–7 (2011)

mmeybodi@aut.ac.ir

A Community Detection Method Based on the Subspace Similarity of Nodes in Complex Networks

Mehrnoush Mohammadi[1], Parham Moradi[1(✉)], and Mahdi Jalili[2]

[1] Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran
m.mohammadi@eng.uok.ac.ir, p.moradi@uok.ac.ir
[2] School of Engineering, RMIT University, Melbourne, Australia
mahdi.jalili@rmit.edu.au

Abstract. Many real-world networks have a topological structure characterized by cohesive groups of vertices. Community detection aims at identifying such groups and plays a critical role in network science. Till now, many community detection methods have been developed in the literature. Most of them require to know the number of communities and the low accuracy in the complex networks are the shortcomings of most of these methods. To tackle these issues in this paper, a novel community detection method called CDNSS is proposed. The proposed method is based on the nodes subspace similarity and includes two main phases; seeding and expansion. In the first phase, seeds are identified using the potential distribution in the local and global similarity space. To compute the similarity between each pair, a specific centrality measure by considering the sparse linear coding and self-expressiveness ability of nodes. Then, the nodes with best focal state are discovered which guarantees the stability of solutions. In the expansion phase, a greedy strategy is used to assign the unlabeled nods to the relevant focal regions. The results of the experiments performed on several real-world and synthetic networks confirm the superiority of the proposed method in comparison with well-known and state-of-the-art community detection methods.

Keywords: Community detection · Sparse mapping · Potential distribution · Label expansion

1 Introduction

Complex networks are important tools for analyzing and studying interactive events in many real-world systems such as biology, sociology and power systems. A common approach to analyze such networks is revealing their hidden community structures while attempting to extract the patterns. In general, a community is considered as a set of nodes having relatively higher inside connections and lower inter-connections. Community detection methods have been applied in many real-world applications such: sentiment analysis [1], recommender systems [2, 3], feature selection [4], skill forming in reinforcement learning agents [5], and link prediction [6]. Community detection methods can be classified into hierarchical [7–9] and partitioning [10]

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 105–120, 2020.
https://doi.org/10.1007/978-3-030-37309-2_9

mmeybodi@aut.ac.ir

methods. Hierarchical methods aim at representing hidden structures of networks as a tree structure [7, 9]. Considering the way of constructing the tree structure, hierarchical methods can be classified into agglomerative and divisive approaches [9]. Agglomerative approaches start from single nodes or initial communities and then merge similar communities through an iterative process until all nodes belong to a single community. On the other hand, divisive approaches assume whole network as a single community and then break down the communities in a repetitive way to form the tree structure. However, requiring a large storage capacity to store the tree structure, and also, finding an appropriate measure for cutting the tree and identifying the community structure are two main issues of these methods. On the other hand, partitioning methods aim at directly grouping the network objects into a set of dens sub-graphs without forming the tree structure [10, 11]. Bisection, spectral clustering, label expansion, and sparse subspace-based methods are the several well-known research lines of divisive community detection methods. Most of them require a high computational cost and thus they cannot efficiently be applied on large scale problems. To address this issue, several evolutionary and nature-inspired methods are proposed to find communities of complex networks. These methods are often divided into single-objective [12], multi-objective [13, 14], and many-objective methods [15]. Single-objective methods aim at discovering communities by optimizing a single-objective function. A majority part of these methods employ the modularity measure in their processes to search through the solution space. This measure computes the number of intra-community edges relative to a null model. Choosing inappropriate quality function may converge the population to identify a sub-optimal result. In other words, most real-world applications require optimizing several competing objectives. On the other hand, multi-objective methods aim at optimizing several quality functions simultaneously to achieve a high-quality result.

Recently, researchers pay much attention to NMF-based methods for identifying communities of networks due to their high interpretability properties. NMF-based methods form an optimization problem to factor the adjacency matrix into two matrices. The former one is membership degrees of nodes to various communities and the latter is the properties of community cores [16]. The objective function is solved by applying the gradient descent method to obtain updating equations the factors [17]. However, requiring the number of communities as primary information, high computational overhead and various results among different runs are the main issues of NMF-based methods. Label expansion methods aim at revealing hidden communities using the topological information of networks [18]. These methods assign labels to some initial nodes as community cores and then propagate the labels among the other nodes through an iterative process. The propagation of labels is continuous until assignment of all of the nodes. However, the random selection of community cores causes the instability of the found solutions. Moreover, the topological information is only considered in the label assignment process and the global information is ignored. In [19] a method a subspace-based community detection method proposed. This method called SSCF first maps the graph to data space by assuming that each node in the graph can be viewed as a linear combination of other nodes laying the same subspace Then, the authors employed a spectral clustering method to identify communities. In another work, the authors of [20] proposed a method called SCE. SCE and

mmeybodi@aut.ac.ir

SSCF both map the graph into a low dimensional space. However, SCE employs a specific label propagation strategy to form final clusters which is much faster than the spectral clustering method. Subspace-based community detection methods use both of the global and topological information of the network in identifying communities and thus they are successful in identifying communities in networks with unclear community structure.

In this paper, we introduce a new Community Detection based on the Nodes Subspace Similarity in the complex networks called CDSNN. This method overcomes to the weakness of the instability of solutions in most of the community detection methods by identifying a definite node of each community. Moreover, in this method, a combination of local and global information is used to identify the boundaries of communities more accurately. To this end, first inspired by [19, 21], the network is mapped to the low-dimension similarity space using the sparse representation technique. So that, in this space each node is shown as a sparse vector of its similarity values to other nodes in its subspace. This information is used to weight the network based on the self-expressive ability of nodes. In the next step, the local maximum nodes in this weighted network are considered as candidate nodes from different communities and, finally, subspace-based label expansion method (SLE) is proposed to expands the focal regions around candidate nodes to borders with the local and global perspective of communities. The main contributions of the proposed method are listed as follows:

1. The proposed method generates stable results in different runs. This is due to identifying the important nodes as community seeds and expanding their labels based on the subspace similarity of nodes. While, sparse subspace-based [19, 21], label propagation [18] and NMF-based methods [17] produce different solutions in different runs due to the random selection of the community centers in their processes.
2. The proposed method identifies the number of communities by discovering a representative node from each community before the label expansion phase. This step can be integrated as a prepossessing step with those which require the number of communities [17, 22].
3. Compared to the several community detection methods such as [18, 22, 23], CDNSS and in general subspace-based methods [24, 25] combine the topological, local and global information using sparse linear coding based on the self-expressiveness ability of nodes.
4. Hierarchical community detection methods require a metric for evaluating the quality of communities [7–9]. Most of these methods employ the modularity metric that causes the low accuracy results on networks with variant sizes of communities. While the proposed method uses a subspace similarity metric which uses both of the local and global information of the networks as well as generating accurate results for various community shapes.
5. The results of the performed experiments on both synthetic and real-world networks based on various qualitative metrics demonstrated the premiere performance of the proposed method in comparison with the traditional and state-of-the-art methods.

mmeybodi@aut.ac.ir

2 Proposed Method

In this paper, a community detection method based on the subspace similarity of nodes called CDNSS (**C**ommunity **D**etection with **N**ode **S**ubspace **S**imilarity) is proposed. The process of the proposed method takes place in two main steps: seeding and expansion. The seeding step aims to identify a proper set of candidate nodes as community seeds. To this end the graph is first mapped to a low dimensional space by hybridizing of a sparse representation technique and a self-expressiveness property of nodes. Then using this representation a novel centrality measure is used to find a set of candidate nodes as community centers. In the expansion step, candidate seeds are expand using a novel label expansion strategy. The key idea behind the label expansion strategy is to expand each candidate seed in such a way to increase the total similarity of nodes within their communities. The overall procedure of the proposed method is exemplified in Fig. 1. Additional details regarding these steps are described in their corresponding section. The details of the proposed CDNSS method is given in Algorithm 1.

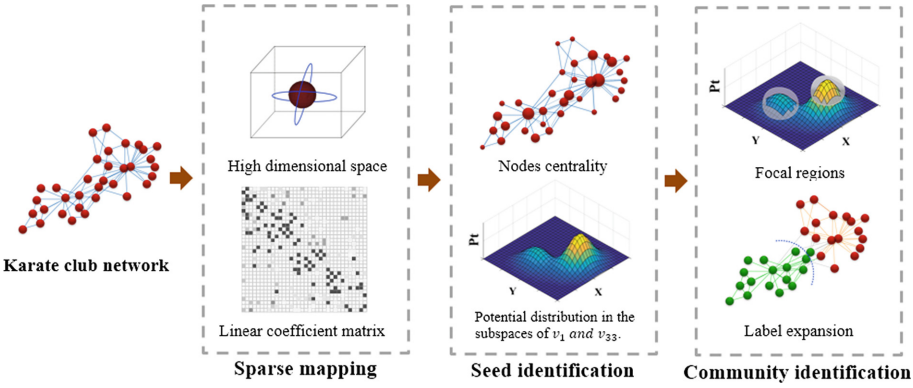


Fig. 1. The flowgraph of the proposed method.

2.1 Phase I: Seeding

The goal of the seeding step is to find a set of candidate nodes as community centers in two steps. The first step maps a network to a low dimensional space using a sparse representation technique and then in the second step, a novel centrality measure is applied on the network nodes to weigh them based on their potential to be as candidate seeds. The details of these steps are described in the following sections.

Sparse Mapping

In this step inspired from [19, 21] the network is mapped to a low dimensional space. To this aim, the following Gaussian kernel is used to map the graph to the similarity space as:

mmeybodi@aut.ac.ir

$$GS_{ij} = \exp\left(-\frac{1}{2}\left(\frac{GD_{ij}}{\sigma_s}\right)^2\right) \tag{1}$$

where GD_{ij} first a is the geodesic distance or the shortest paths between v_i and v_j, δ is a decay rate which is set by a constant value. This kernel focuses on the distribution of data in the similarity space and can be considered as a non-linear function of geodesic distance that is bounded between 0 and 1. The next step is to locate a high-dimensional data into a lower-dimensional unit by using the sparse representation method proposed in [26]. In other words the sparse representation technique is combined with a self-expressiveness ability of nodes to reduce the dimensionality. Considering the self-expressiveness ability of nodes, each node in the similarity space can be expressed as a linear combination of the other nodes. This property can be formulated as:

$$GS_i = \sum_{j=1..n, j\neq i} \gamma_{ij} GS_j \tag{2}$$

where γ_{ij} denotes the similarity weights between nodes i and j. The aim is to find absolute and disjoint subspaces in such a way that can be satisfied with the following objective function:

$$\gamma_i^* := \arg\min_{\gamma_i} \left\| GS_i - \widehat{GS}\gamma_i \right\|_2^2 + \lambda \|\gamma_i\|_1 \tag{3}$$

where GS_i refers to the i-th column of GS, $\widehat{GS} = GS \backslash GS_i$, $\|\cdot\|_1$ is Manhattan norm or l_1 norm that is used to control the sparsity of coefficient similarity vectors [26–28] and λ is a parameter that controls the sparsity of the coefficient. Based on this observation, the problem is turned into a convex optimization problem and can be solved using convex programming frameworks [29]. Here we used the ADMM [30] to solve the objective function of Eq. (3). Afterward the similarity weights are normalized by $\Gamma_{ij} = \frac{1}{2}(\gamma_{ij} + \gamma_{ji})$. Considering the self-expressiveness ability of nodes we propose a novel centrality measure to rank the nodes based on their importance as:

$$R(v_i) = \gamma_i GS_i^T \tag{4}$$

Seed Identification

This step aims to locate a set of representative seeds as community cores. The idea is to identify those nodes with maximum potential in their influence region and introduce them as community seeds. The influence region of node v_i is determined as:

$$IR(v_i) = \{v_j | \forall j = 1...n, j \neq i, \Gamma_{ij} > \alpha\} \tag{5}$$

where α is the threshold that controls the extent of the influenced region for each node. In this paper we choose those nodes that have maximum potential value in their influence region as candidate seeds. So, each candidate seed is a representative of a

mmeybodi@aut.ac.ir

different community. While seeds have local maximum-potential value, thus they are supposed to be located in the cores of dense areas. This strategy can also be used as a pre-processing step to determine the number communities.

2.2 Phase II: Expansion

In this section, we proposed an effective subspace-based label expansion method (SLE) to form communities around identified community seeds. The proposed SLE method first, forms focal regions around the seeds and then a greedy strategy is adopted to add those of the unlabeled nodes to their closest region. Each focal region is a set of nodes where their similarity to the community core are higher than a threshold value and it is defined as:

$$\mathrm{FR}(\mathrm{C}_i) = \{\mathrm{v_j} | \forall \mathrm{j} = 1 \ldots \mathrm{n}, \mathrm{j} \neq \mathrm{C_i}, \Gamma_{j,C_i} > \beta\} \cup \mathrm{C_i} \tag{6}$$

where β is a threshold value that is used to control the subspace density of focal regions and its higher value leads to form denser canonical regions. After the formation of focal regions, the next step is to assign those of unlabeled nodes to the closest regions with aims to maximize their densities. In other words to assign an unlabeled node to region or a primary identified community, its similarity to all members of the region is computed. The node is assigned to a region that its total similarity value is higher than the other primary information. Here we use the following equation to compute the similarity between each pair as:

$$sim_{i,j} = \gamma_i \cdot \gamma'_j \tag{7}$$

3 Results

In this section, the proposed method is compared with several well-known and state-of-the-art community detection methods on both real-world and synthetic networks. To this end, two validity metrics i.e., Normalized Mutual Information and Coverage are used to evaluate the performance of methods.

3.1 Networks

In this paper, several networks with different properties are used in the experiments to show the performance of our algorithm. In these experiments, we use two common types of benchmark networks which have the most use in community detection methods: synthetic and real-world networks.

Synthetic Networks: The most realistic feature of the Synthetic networks used is their compliance with the power-law degree distribution in their nodes degree and communities size. This model is generated by Lancichinetti, Fortunato, and Radichi which called LFR benchmark networks and able to generate the networks with implanted communities within them [31]. The source code of this model is available on the

mmeybodi@aut.ac.ir

https://sites.google.com/site/andrealancichinetti/Home. The details of the adjustable parameters in the LFR model are summarized in Table 1.

Real-World Networks: To makes our experiments more realistic, we used many real-world networks including, Karate club, Dolphins, US Political Books, Email_EU_core Jazz and E-coli networks which the community structure is clear in some of them. Their details are described in Tables 2 and 3.

Algorithm 1. CDNSS: Community Detection with Node Subspace Similarity

Input *A: Adjacency matrix of network.*
α: *To control the extent of the influenced regions.*
β: *To control the subspace density of focal regions.*

Output *Communities*

Begin algorithm

Phase I: Seeding

seeds =[];
Step 1: Sparse mapping
GS ← *Mapping graph to the similarity space using Eq.1.*
γ ← *Dimension reduction of similarity space using Eq.2*
$\Gamma \leftarrow \frac{1}{2}(\gamma + \gamma^{T})$;
R ← *Calculate the potential of nodes using Eq.3.*
Step 2: Seed identification
IR ← *Identify the influence region for nodes using Eq.4.*
For i=1 to N
If node_i *has the maximum R in its influence region*
seeds ← *[seeds* node_i*].*
End if
End for

Phase 2: Expansion

FR ← *Form focal regions around seeds using Eq.5.*
For i=1:number of unlabeled nodes
Max_sim ← *0.*
For j=1:number of focal regions
sim ← *calculate the similarity of* unlabeled_node_i *to the members of the* focal_region_j *using Eq.6.*
If sim > max_sim
Max_sim ← *sim.*
Index ← *j.*
End if
End for
Assign unlabeled_node_i *to the* $\text{focal_region}_{\text{index}}$.
End for
Communities ← *focal regions.*
End of the algorithm

mmeybodi@aut.ac.ir

Table 1. Adjustable parameters in the LFR benchmark model.

Notations	Description
N	The number of nodes in the network
|E|	The number of edges in the network
M	Mixing parameter: Degree of connection between communities
K	The average degree of nodes
maxK	The maximum degree of nodes in the network
minC	The minimum size of communities
maxC	The maximum size of communities

Table 2. Real-world networks used in the experiments. N, |E| and C show the numbers of nodes, edges and communities, respectively, and k is the averaged degree of nodes.

Networks	N	|E|	k	C	Description
Karate	34	78	4.59	2	The relationships between karate club members in 1977
Dolphins	62	159	5.13	2	The repeated associations between dolphins in Doubtful Sound, New Zealand [32]
Polbooks	105	441	8.40	3	A network of US political books diffused in the 2004 presidential choice
Jazz	198	2742	27.69	–	A Jazz musicians collaboration network [33]
E-coli	329	456	2.77	–	The transcriptional regulation network of Escherichia coli [34]
Email	986	16064	32.58	42	A network of incoming emails from a European research establishment

Table 3. Details of benchmark networks with $\mu = 0.7$. n, E, K, maxK, minC = 20, maxC = 50, and NGTC are the number of nodes, number of edges, average degree, maximum degree, minimum size of communities, maximum size of communities and the number of ground-truth communities, respectively. σ_s is decay rat in the Gaussian similarity function.

Networks	Features					
	N	E	k_m	max_K	NGTC	σ_s
Net1	700	3782	15	20		1.0712
Net2	1000	7631	15	20		1.0634
Net3	1500	11567	15	20		1.0687
Net4	2000	31000	15	20		1.0889

3.2 Performance Metrics

The existence of evaluation metrics is essential to verify the performance of community detection methods and also a comparison of them. In practice, these metrics grouped into two categories: information recovery and qualitative metrics.

Information Recovery Metrics: this metrics compare the standard class partition of networks with the partition obtained from community detection methods. Normalized

mmeybodi@aut.ac.ir

Mutual Information (NMI) [35] is a famous measure in this category used in this paper. Let A be the ground truth communities structure and B be the communities structure obtained from community detection methods.

Normalized Mutual Information (NMI) [35] is based on the information theory [36] can be formulated as follows:

$$\mathrm{NMI}(A,B) = \frac{\sum_{ij} n_{ij}\log\left(\frac{n_{ij}}{n_i^A n_j^B}\right)}{\left(\sum_i n_i^A \log(n_i^A)\right) + \left(\left(\sum_j n_j^B \log(n_j^B)\right)\right)} \tag{9}$$

Where n_{ij} denotes the number of agreements between community i and j in partitions A and B respectively. n_i^A and n_j^B are the number of nodes in the community i in the partition A and community j in partition B respectively [37].

Qualitative Metrics: qualitative metrics such as coverage are based on the calculation of the quality of communities accessed from community detection methods and do not demand to know the community structures. Various approaches have been used to measure the quality of communities, for instance, community quality is defined as the ratio of the number of intra communities edges to all of the edges in the network, in the coverage metric [38].

3.3 Comparison Methods

In experiments, several well-known and state-of-the-art methods are employed to the comparison that a brief introduction of each as follows:

- **FN** [8] is based on the bottom-up approach that grouped into hierarchical-agglomerative methods. FN uses the modularity metrics to merging subgraphs in each iteration.
- **GN** [9] is in the category of hierarchical-divisive methods that it uses the betweenness and modularity metrics to split and evaluate the network in each iteration.
- **LPA** [18] is the most popular label propagation method that used only the structure-property of networks to identify communities. So, the label of each node is updated using the label of the maximum number of its neighbor in each iteration.
- **LUV** [39] is the heuristic method based on the modularity optimization.
- **Infomap (Info)** [40]. In this method, communities are discovered with aims to minimize the expected description length of a random walker path.
- **Walktrap (WT)** [41] is grouped into hierarchical methods which use the random walk strategy to evaluate and merge communities structure.
- **LE** [42] uses the non-negative eigenvector of modularity matrix to discover communities structure in the complex networks.
- **B. Saoud (MSP)** [7] is in the category of hierarchical-agglomerative methods that uses the Modularity and spanning tree of nodes dissimilarity to form communities.
- **SSCF** [19] is one of the popular subspace-based methods that it uses the self-expression ability of nodes to create an affinity matrix in the similarity space. Then the spectral clustering method is applied to this matrix to explore final communities

mmeybodi@aut.ac.ir

- **X. Tang et al. (TNMF)** [17] is based on the NMF model with both local and global perspective of the network. In this method, Jaccard similarity and Page Ranke personality methods are used to calculate local and global information respectively.

3.4 Parameter Settings

There are two regularity parameters with different roles in the CDNSS (i.e. α and β). α controls the locality of communities. So, as the α get closer to one, the concept of community is more local. And the subspace of nodes is more limited. As a result, the number of identified communities by CDNSS also increases. The proper range of α on the Karate club and Dolphins networks is [0 0.05]. While Fig. 6(b) is shown the [0.05 0.06] as suitable range of α on the US Political Books. Therefore $\alpha = 0.05$ is the appropriate value to discovering the correct number of communities in the real-world networks. The parameter β controls the similarity density within the focal regions. In the next experiment, the effect of both α and β are investigated on the performance of the CDNSS in the $\alpha \in [00.3]$ and $\beta \in [00.3]$. To this end first, CDNSS method is performed on the Karate club, Dolphins and US Political Books networks. In this experiment, β is set to zero and $\alpha \in [00.3]$ and the results are evaluated based on the NMI. The previous results show the [0 0.05) as suitable range of α on the tested real-world networks. Also, this fig indicates the importance of α on the performance of CDNSS. Then the proper values for β are found by set $\alpha = 0$ and $\beta \in [00.3]$. Fig. 8 shows that as $\beta \in [0.020.06]$ CDNSS has the best performance on all three real-world networks.

LPA [18], SSCF [19] and X. Tang et al. [17] methods are used the randomness factor in their process that in our experiments the average results over 100 independent runs are reported. As well as, X. Tang et al method is based on the NMF model which it uses the gradient descent approach to solve the final model. There are two stop conditions in this model i.e. error rate parameter (i.e. ε) and the maximum number of iteration (i.e. Maxiter) which are set to values 10^{-4} and $2 * 10^3$, respectively. In this paper, the iGraph package in the R programming language is used to run FN [8], GN [9], LPA [18], LUV [39], Infomap [40], Walktrap [41] and LE [42]. Also, MSP, SSCF, T(NMF) and CDNSS methods are implemented in MATLAB 2016. All of community detection methods have run on a computer with Core i5 CPU and 8 GB RAM.

3.5 Experiment Process and Results

The aims of this section is to compare the power of different community detection methods in discovering communities on both synthetic and real-world networks. To this end, community detection methods are compared in terms of NMI and coverage in the separate subsection.

Test on Synthetic Networks

In this section, to prove the power of the CDNSS for discovering communities in the complex network, several LFR networks with the different property are used in two experiments. First, the performance of the SSCF and CDNSS methods are compared using NMI metric on the LFR networks with $N = 100$, $k = 10$, $MinC = 6$, $MaxC = 30$ and $\mu \in [0.10.8]$. As specified, the difference between these networks is the clarity

mmeybodi@aut.ac.ir

of communities structure which controls by μ. The results of this experiment are shown in Fig. 2. As is evident from it, both methods have high performance in identifying communities in the networks with $\mu \in \{0.1, 0.2, 0.3, 0.4\}$ which are close to one. But the complex structure of communities in Networks with $\mu \in \{0.6, 0.7, 0.8\}$ has led to low accuracy in both methods. However, the superiority of CDNSS is clear in most cases, as special in the networks with $\mu = 0.8$ SSCF method has the performance close to zeros (i.e. NMI = 0.0966) while CDNSS has much better performance(i.e. NMI = 0.2581). Figure 3 represent networks used with $\mu = 0.2$ and $\mu = 0.7$.

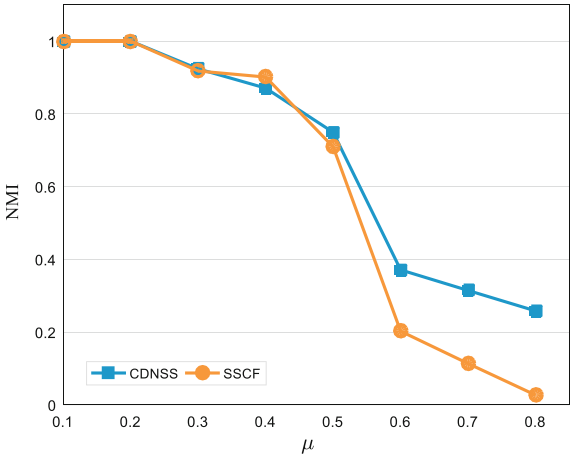


Fig. 2. Validate of SSCF and CDNSS methods in terms of NMI on the LFR networks with N = 100, k = 10, minC = 6, maxC = 30 and $\mu \in [0.1\ 0.8]$.

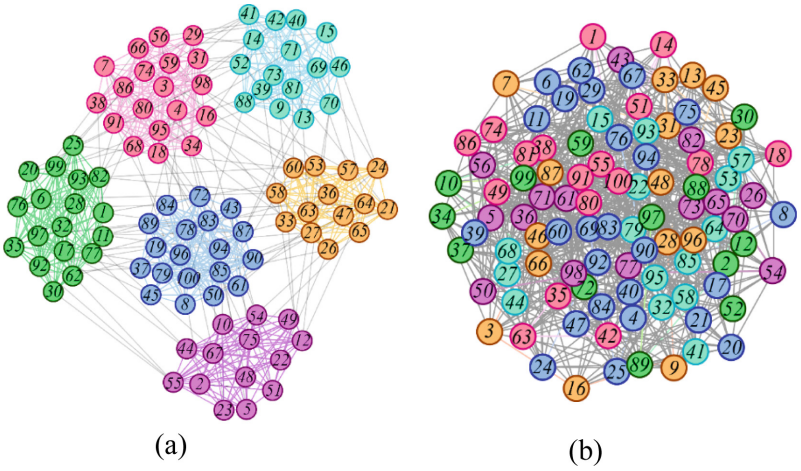


Fig. 3. Clarity of communities' structure in the LFR networks with (a) $\mu = 0.2$ and (b) $\mu = 0.7$.

mmeybodi@aut.ac.ir

Most of the community detection methods are unable to discover communities in the networks with $\mu > 0.7$. So, these networks are a big challenge for community detection methods. In the next experiment, Net1, Net2, Net3, and Net4 are used to compare the performance of community detection methods. Figure 4 shows the results of NMI metrics obtained from different methods on these networks, respectively. LPA and Infomap methods have weak performance and close to zero in these networks. So, their results are not reported. Also, GN method has a high complexity time.

As shown in Fig. 4, the proposed method has the best performance on the Net1, Net2, and Net3 in term of NMI. While in network 4, it is in second place after SSCF. However, the average performance of the CDNSS is 0.5620 and has the first rank among the tested methods on the Net1, Net2, Net3, and Net3. SSCF and MSP methods are in the second and third place, respectively. These results demonstrate the ability of a CDNSS in discovering communities in complex networks.

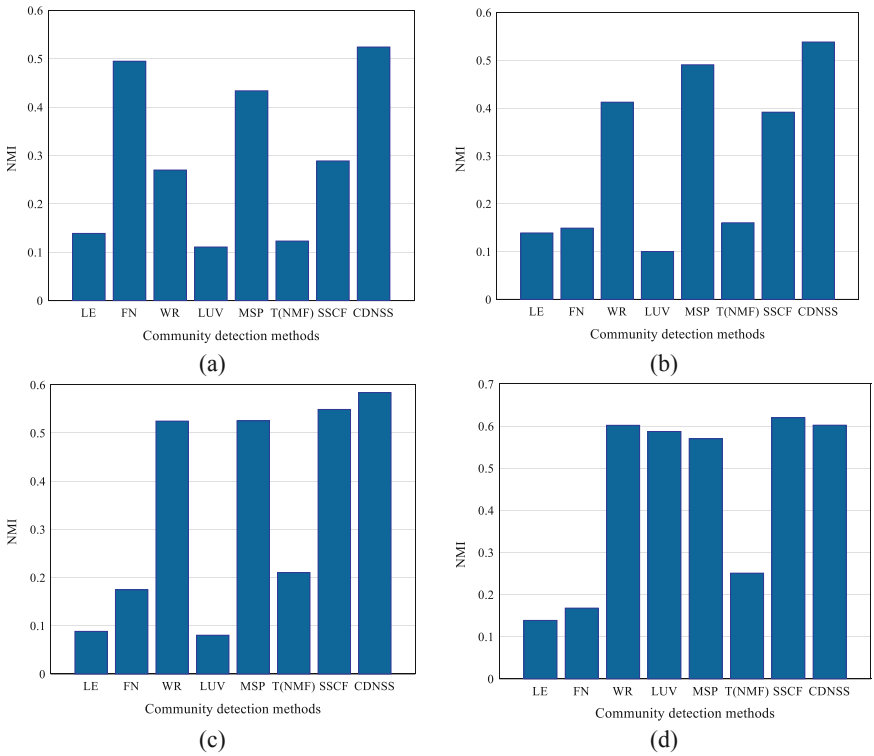


Fig. 4. Comparison of different community detection methods on the, (a) Net1, (b) Net2, (c) Net3 and, (d) Net4 in term of NMI.

mmeybodi@aut.ac.ir

Test on the Teal-World Networks

In this section for more realistic experiments, several real-world networks with the known and unknown structure of communities are tested. Table 4 represents the numerical results obtained from different community detection methods on the networks with ground-truth communities (i.e. Karate Club, Dolphins, US Political Books, and Email-EU-core). US Political Books and Email-EU-core networks have a more complex structure than other networks. So that the low performance of the community detection methods in these networks is an affirmation of this claim. The results in theses tables show that the CDNSS has more acceptable performance over other methods on these networks. Also, CDNSS methods have the best performance in the Dolphins network. While CDNSS method has the second rank performance on the Karate club network. Numerical results of the average rank of methods in tested networks confirm the superiority of the CDNSS method among other methods on the real-world networks. Also, in this table the rank of methods among others are designated next to their performance. In this section, another experiment is performed to evaluate the performance of the community detection methods on the real-world networks with the known and unknown structure of communities. To this end, the quality of discovered communities is evaluated based on the Coverage metric. Also, X. Tang et al method need to know the number of communities in the networks. For this purpose, FN method is used to detect the number of communities in the networks with unknown structure of communities. Numerical results of this experiment are reported in Table 5. These results indicate that the discovered communities by CDNSS have the most or second-most quality than other competing community detection methods on the real-world networks with known and unknown community structure.

Table 4. NMI results on the Karate Club, Dolphins, US Political Books, and Email-EU-core networks. #AR shows the average rank of methods on the tested real-world networks.

Methods	Networks				
	#AR	Karate	Dolphins	Books	Email
GN	4	0.836/3	0.751/4	0.558/5	0.599/4
LE	8.25	0.677/7	0.130/10	0.520/8	0.504/8
FN	6.5	0.692/6	0.557/5	0.530/6	0.427/9
Info	6.5	0.699/4	0.131/9	0.269/10	0.610/3
WT	8.75	0.504/10	0.131/9	0.283/9	0.518/7
LUV	6.75	0.670/8	0.488/6	0.526/7	0.536/6
MSP	5.5	0.602/9	0.438/7	0.583/4	0.628/2
LPA	10.25	0.396/11	0.132/8	0.112/11	–
T(NMF)	4.25	**1/1**	0.767/3	0.590/3	0.265/10
SSCF	3.25	0.785/4	**0.881/2**	**0.618/2**	0.596/5
CDNSS	1.25	**0.837/2**	**1/1**	**0.677/1**	**0.629/1**

mmeybodi@aut.ac.ir

Table 5. Numerical results obtained from different community detection methods on the Karate Club, Dolphins, US Political Books, Jazz, E-coli and Email-EU-core networks in term of coverage.

Methods	Networks					
	Karate	Dolphin	Books	Jazz	E-coli	Email
GN	0.832/3	0.887/4	0.905/3	0.709/8	0.864/4	0.367/7
LE	0.667/9	0.547/9	0.778/7	0.771/6	0.811/10	0.531/5
FN	0.756/5	0.824/5	0.918/2	0.779/5	0.853/6	0.685/2
Info	0.821/4	0.695/7	0.397/9	0.139/11	0.743/11	0.106/10
WT	0.590/10	0.695/7	0.580/8	0.789/4	0.866/3	0.679/3
LUV	0.731/6	0.767/6	0.891/4	0.732/7	0.860/5	0.617/4
MSP	0.679/8	0.654/8	0.880/6	0.612/9	0.830/9	0.403/6
LPA	0.718/7	0.465/10	0.315/10	0.903/2	0.835/7	–
T(NMF)	**0.872/1**	0.927/3	0.882/5	0.535/10	0.831/8	0.188/9
SSCF	0.821/4	0.956/2	0.880/6	0.795/3	0.902/2	0.366/8
CDNSS	0.859/2	**0.962/1**	**0.943/1**	**0.921/1**	**0.943/1**	**0.775/1**

4 Conclusion

In this paper a community detection method called CDNSS is proposed based on the subspace similarity of nodes is the network. The aim of CDNSS is to identify important nodes in the network and then forming communities using a label propagation method. These are done in the two main phases; seeding and expansion. In the former phase, a novel centrality measure is used to rank the nodes based on their importance. In the second phase, a greedy strategy is used to discover the most prominent nodes in each community. The communities are formed around the core nodes by hybridization of local and global perspective. Experimental results on the synthetic and real-world networks confirm the superiority of CDNSS among other community detection methods in terms of qualitative and information recovery metrics.

References

1. Eliacik, A.B., Erdogan, N.: Influential user weighted sentiment analysis on topic based microblogging community. Expert Syst. Appl. **92**, 403–418 (2018)
2. Moradi, P., Ahmadian, S., Akhlaghian, F.: An effective trust-based recommendation method using a novel graph clustering algorithm. Phys. A **436**, 462–481 (2015)
3. Rezaeimehr, F., Moradi, P., Ahmadian, S., Qader, N.N., Jalili, M.: TCARS: time- and community-aware recommendation system. Future Gener. Comput. Syst. **78**, 419–429 (2018)
4. Moradi, P., Rostami, M.: Integration of graph clustering with ant colony optimization for feature selection. Knowl.-Based Syst. **84**, 144–161 (2015)
5. Rad, A.A., Hasler, M., Moradi, P.: Automatic skill acquisition in reinforcement learning using connection graph stability centrality. In: 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 697–700 (2010)

mmeybodi@aut.ac.ir

6. Wang, Z., Wu, Y., Li, Q., Jin, F., Xiong, W.: Link prediction based on hyperbolic mapping with community structure for complex networks. Phys. A Stat. Mech. Appl. **450**, 609–623 (2016)
7. Saoud, B., Moussaoui, A.: Community detection in networks based on minimum spanning tree and modularity. Phys. A Stat. Mech. Appl. **460**, 230–234 (2016)
8. Newman, M.E.: Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**, 066133 (2004)
9. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**, 026113 (2004)
10. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**, 75–174 (2010)
11. Capocci, A., Servedio, V.D., Caldarelli, G., Colaiori, F.: Detecting communities in large networks. Phys. A **352**, 669–676 (2005)
12. Moradi, M., Parsa, S.: An evolutionary method for community detection using a novel local search strategy. Phys. A **523**, 457–475 (2019)
13. Ghaffaripour, Z., Abdollahpouri, A., Moradi, P.: A multi-objective genetic algorithm for community detection in weighted networks. In: 2016 Eighth International Conference on Information and Knowledge Technology (IKT), pp. 193–199 (2016)
14. Rahimi, S., Abdollahpouri, A., Moradi, P.: A multi-objective particle swarm optimization algorithm for community detection in complex networks. Swarm Evol. Comput. **39**, 297–309 (2018)
15. Tahmasebi, S., Moradi, P., Ghodsi, S., Abdollahpouri, A.: An ideal point based many-objective optimization for community detection of complex networks. Inf. Sci. **502**, 125–145 (2019)
16. Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized nonnegative matrix factorization for data representation. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 1548–1560 (2011)
17. Tang, X., Xu, T., Feng, X., Yang, G., Wang, J., Li, Q., Liu, Y., Wang, X.: Learning community structures: global and local perspectives. Neurocomputing **239**, 249–256 (2017)
18. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**, 036106 (2007)
19. Mahmood, A., Small, M.: Subspace based network community detection using sparse linear coding. IEEE Trans. Knowl. Data Eng. **28**, 801–812 (2016)
20. Mohammadi, M., Moradi, P., Jalili, M.: SCE: subspace-based core expansion method for community detection in complex networks. Phys. A **527**, 121084 (2019)
21. Tian, B., Li, W.: Community detection method based on mixed-norm sparse subspace clustering. Neurocomputing (2017)
22. Wang, F., Li, T., Wang, X., Zhu, S., Ding, C.: Community discovery using nonnegative matrix factorization. Data Min. Knowl. Disc. **22**, 493–521 (2011)
23. Chen, Z., Xie, Z., Zhang, Q.: Community detection based on local topological information and its application in power grid. Neurocomputing **170**, 384–392 (2015)
24. Tian, B., Li, W.: Community detection method based on mixed-norm sparse subspace clustering. Neurocomputing **275**, 2150–2161 (2018)
25. Mahmood, A., Small, M.: Subspace based network community detection using sparse linear coding. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), pp. 1502–1503. IEEE (2016)
26. Elhamifar, E., Vidal, R.: Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 2765–2781 (2013)
27. Xu, J., Xu, K., Chen, K., Ruan, J.: Reweighted sparse subspace clustering. Comput. Vis. Image Underst. **138**, 25–37 (2015)
28. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, pp. 849–856 (2002)

mmeybodi@aut.ac.ir

29. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
30. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. **3**, 1–122 (2011)
31. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E **78**, 046110 (2008)
32. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav. Ecol. Sociobiol. **54**, 396–405 (2003)
33. Gleiser, P., Danon, L.: Community structure in jazz. Adv. Complex Syst. **6**, 565 (2003)
34. Shen-Orr, S.S., Milo, R., Mangan, S., Alon, U.: Network motifs in the transcriptional regulation network of Escherichia coli. Nat. Genet. **31**, 64 (2002)
35. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
36. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**, 2837–2854 (2010)
37. Zhang, Z.-Y., Wang, Y., Ahn, Y.-Y.: Overlapping community detection in complex networks using symmetric binary matrix factorization. Phys. Rev. E **87**, 062803 (2013)
38. Kobourov, S.G., Pupyrev, S., Simonetto, P.: Visualizing graphs as maps with contiguous regions. In: EuroVis 2014, Accepted to appear (2014)
39. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech: Theory Exp. **2008**, P10008 (2008)
40. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proc. Natl. Acad. Sci. **105**, 1118–1123 (2008)
41. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: International Symposium on Computer and Information Sciences, pp. 284–293. Springer (2005)
42. Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**, 036104 (2006)

mmeybodi@aut.ac.ir



Forecasting Multivariate Time-Series Data Using LSTM and Mini-Batches

Athar Khodabakhsh[1(✉)], Ismail Ari[1], Mustafa Bakır[2], and Serhat Murat Alagoz[2]

[1] Department of Computer Science, Özyeğin University, Istanbul, Turkey
athar.khodabakhsh@ozu.edu.tr, ismail.ari@ozyegin.edu.tr
[2] Software Development Department , TÜPRAŞ, Kocaeli, Turkey
{mustafa.bakir,serhatmurat.alagoz}@tupras.com.tr

Abstract. Multivariate time-series data forecasting is a challenging task due to nonlinear interdependencies in complex industrial systems. It is crucial to model these dependencies automatically using the ability of neural networks to learn features by extraction of spatial relationships. In this paper, we converted non-spatial multivariate time-series data into a time-space format and used Recurrent Neural Networks (RNNs) which are building blocks of Long Short-Term Memory (LSTM) networks for sequential analysis of multi-attribute industrial data for future predictions. We compared the effect of mini-batch length and attribute numbers on prediction accuracy and found the importance of spatio-temporal locality for detecting patterns using LSTM.

Keywords: LSTM · Multivariate time-series · RNN · Sensors · Sequence data · Time-series

1 Introduction

Industrial IoT (IIoT) devices collect data from complex physical devices and instruments that have time-varying and nonlinear behavior. Forecasting the future is a challenging task which is possible by analysis of short and long-term dependencies on data. Furthermore, predictions are more accurate when the dependencies between variables are better modeled [1]. In learning methods, we desire the models to learn dependencies automatically by observing the past data to predict the future. These methods are gaining attention for industrial applications in training nonlinear models in large dimensions over fast flowing data and large historical datasets. RNNs and LSTM are now proven to be effective in processing time-series data for prediction [2].

For multivariate time-series prediction, several Deep Learning architectures are used in different domains such as stock price forecasting [3], object and action classification in video processing [4], weather and extreme event forecasts [5]. In many applications, the high-dimensional data has high correlation among dimensions and these correlations are spatially located close to each other that

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 121–129, 2020.
https://doi.org/10.1007/978-3-030-37309-2_10

mmeybodi@aut.ac.ir

consequently get reflected in deep neural networks for local processing [6]. For non-spatial data like time-series, the relationship and correlations among measurements can be exploited by sequence analysis which is traditionally applied by sliding-window approach. Industrial applications of these analyses can be fault detection [7], automated control, and predictive maintenance [8].

In all industries including Oil & Gas, there is a need to forecast input (*e.g.* crude oil) supply needs, depending on the current output (*e.g.* gasoline, diesel, *etc.*) demands. Refineries can make future contracts based on analysis results to reduce their uncertainties. In these mission critical businesses, thousands of sensors are installed around physical equipment and Supervisory Control and Data Acquisition (SCADA) systems measure flow, pressure, temperature of turbines, pumps, and injectors. Achieving continuous safety, process efficiency, long-term durability, and planned (vs. unplanned) downtimes are among the main goals for industrial plant management. These controls and actions should be performed in real-time according to temporal patterns received from stream data.

Since most industrial systems are dynamic and the relation among variables are complex, dynamic, and nonlinear, the quality of models and predictions are dependent on the current context of the system [9]. Therefore, LSTM can be used for sequence processing over time-series data, depending on the historical and current context. In this paper, we used time-series data from the petrochemical plant of a real oil refinery with approximately 11.5 million $ton/year$ processing capacity [10].

2 Background and Related Work

Analysis of time-series data has been a subject of interest for scientific and industrial studies. They are used for knowledge extraction, prediction, classification, and modeling of time-varying systems. Depending on the context, different linear and nonlinear modeling techniques are applicable on data. Linear models such as Auto Regressive Moving Average (ARMA) [11] make short-term predictions, but extracting long-term dependencies are also demanded while mining historical data. Utilizing NNs and networks with memory such as RNNs and LSTM provides the ability to process temporal patterns in addition to long-term dependencies. Lai *et al.* [1] proposed a novel framework called LSTNet that uses the Convolutional Neural Network (CNN) and RNN to extract short-term local dependency patterns among variables and to discover long-term patterns for time-series trends. Jiang *et al.* [3] used RNNs and LSTM for time-series prediction of stock prices. Loganathan *et al.* [12] used LSTM for multi-attribute sequence-to-sequence (Seq2Seq) model for anomaly detection in network traffic. Gross *et al.* [6] interpreted time-series as space-time data for power price prediction. In our previous work [13], we used ARMA for modeling the short-term dependencies of attributes for error detection and in this study, we investigate the effect of long-term dependencies on prediction to improve our models for multi-mode analysis in real-time.

mmeybodi@aut.ac.ir

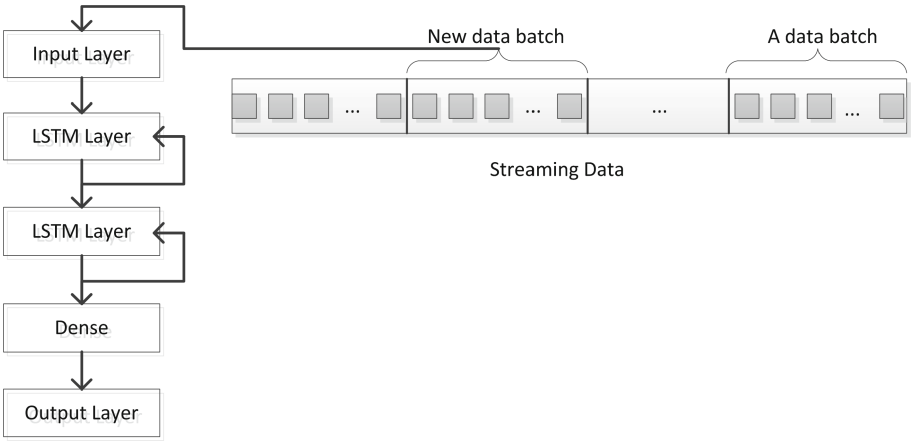


Fig. 1. Stacked architecture of LSTM networks used for supply prediction. The time-series data are transformed into spatial data in mini-batches that consist of multivariate sensor data in each box.

3 Methodology

For capturing the dependencies and extracting long-term patterns in time-series data, it is beneficial to use stacked LSTM networks. The relation among attributes change over time and it is important to react to this change to update the model. The challenge is to decide how many steps to look back into prior data. In most of the recent studies, the focus is on the neural network structure whereas in this study we investigate the effect of memory size and importance of local sequence analysis on training the network and prediction accuracy of future values. In our previous study [14], we managed to identify operational modes by detecting the changing patterns observed in time-varying systems.

3.1 Problem Formulation to Define and Fit LSTM

As shown in Fig. 1, we converted non-spatial multivariate time-series sensor data into time-space frames (similar to pictures in a movie) and trained model for sequence prediction of industrial sensor data using LSTM. Each mini-batch consists of multivariate sensor data that is received consecutively. Rows of data are then transformed to a time-space frame by adding current data to the sequence in a given time on top of prior data building the mini-batches. The learning network consists of two LSTM layers and one *Dense* layer. This network is then used for unsupervised modeling which can learn long-term correlated variables.

For multivariate time-series forecasting, given the series $X = \{x_1, x_2, \ldots x_{t-1}\}$, where x_i represents values at time i, the task is to predict value of x_t. For predictions we used $\{x_{t-w}, x_{t-w+1}, \ldots x_{t-1}\}$ where w is the window size. These sequences of mini-batches are then fed into a two-layer LSTM network in n epochs for training and for p step ahead predictions. The dataset is split into training and testing sets. The network is trained with Adam backpropagation on mini-batches.

mmeybodi@aut.ac.ir

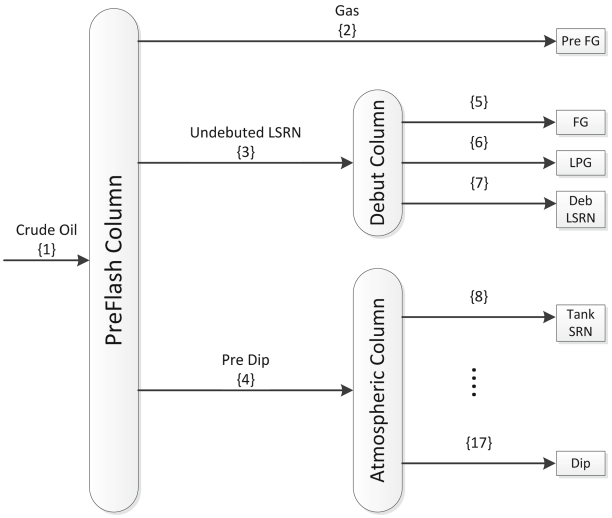


Fig. 2. A simplified petrochemical plant model showing columns for processing crude oil and other by-products.

3.2 Tuning LSTM Hyperparameters

Hyperparameters determine many NN features such as the network structure, dropout, learning rate, and activation functions. It is challenging to optimize both the batch-sizes and hyperparameters. We ran a sensitivity analysis over several of these dimensions for stacked LSTM behavior including different activation functions and number of neurons and evaluated the Root Mean Squared Error (RMSE) values for selecting the appropriate hyperparameters to be used for forecasting.

4 Experiments and Results

4.1 Petrochemical Plant Case Study

For demonstration, we obtained time-series data from a real petrochemical plant and applied stacked layers of LSTM for predicting crude oil purchase amount. Depicted in Fig. 2, a simplified plant model that has 17 flow sensors over 3 main branches of material flows and the corresponding sensor data streams. Crude oil columns take the oil as input and deliver several by-products such as liquid propane gas, fuel oil, kerosene, diesel, and asphalt. A preflash unit reduces the pressure and provides vaporization, where the vapor goes to a debutanizer for distillation and the liquid mix goes to an atmospheric column for separation. This time-series data is time-framed such that the measurements at current time t is predicted by given measurements of the by-products from the prior time step.

mmeybodi@aut.ac.ir

In Fig. 3 a fraction of crude oil data is depicted that is used for training the LSTM network. This dataset contains flow rates of crude oil measurement as input and outputs of the plants for processed by-products of 3 main branches of the petrochemical plant.

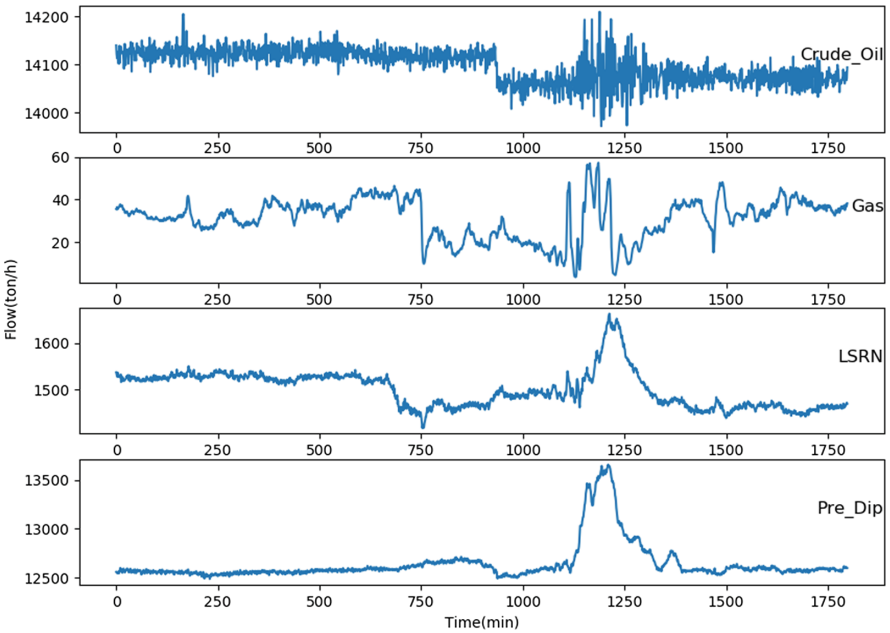


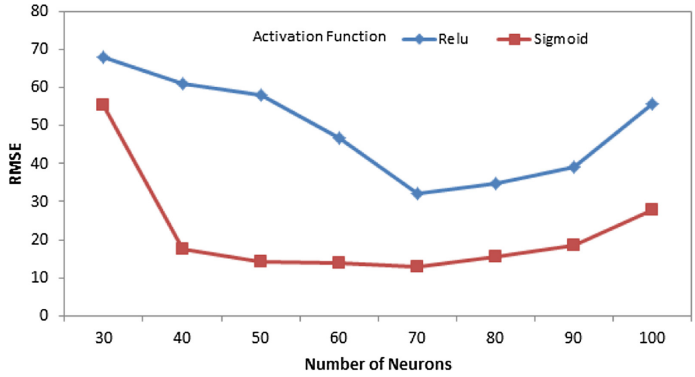
Fig. 3. Flow rates (ton/h) of Crude Oil and three main branches of by-products including Propane Gas, LSRN and, Pre Dip that show correlated and dynamic behavior of Petrochemical plant's production.

4.2 Experimental Results

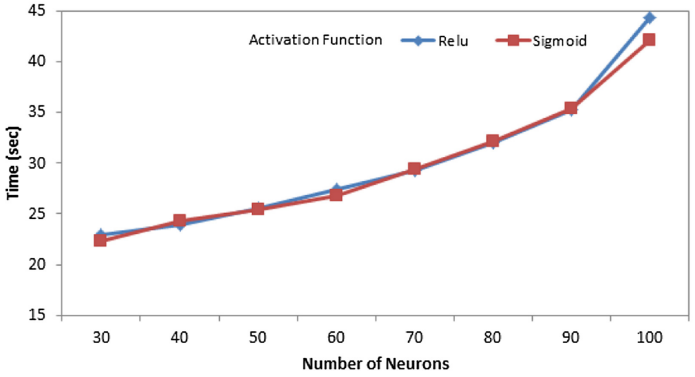
We trained the LSTM on the multivariate data for time-series forecasting using Tensorflow [15] in Python with Keras. The model is trained over 6 days measurements and tested over 30 min of data and RMSE values are computed for evaluating the accuracy of the model and predicted values. The Mean Square Error (MSE) loss function and the efficient Adam version of stochastic gradient descent [16] optimization is used in the LSTM network. The first LSTM layer, as shown in stacked architecture in Fig. 1, is trained and the output of this sequence analysis is fed into the second layer of the network which is another LSTM layer. The input shape is 1 time step with 2, 7, and 17 attributes. The model is applied for 50 training epochs with different batch sizes for comparison of prediction accuracy. The loss of train and test steps are evaluated for the validation data during model training. After fitting the model, the forecasted

mmeybodi@aut.ac.ir

values are obtained for test dataset. Comparing the forecasted and actual values in the original scale, the RMSE value of the model is calculated.



(a)



(b)

Fig. 4. Effect of number of neurons on (a) RMSE value, (b) computation time for training the LSTM network, using Relu and Sigmoid activation functions.

For NN configuration, a hyperparameter tuning approach is applied to extract best parameters to improve the accuracy. We evaluated the behavior of LSTM network for different activation functions with respect to the number of neurons; the parameters that minimize the RMSE are later selected. We compared the effects of *ReLU* (Rectified Linear Unit) and *Sigmoid* activation functions and as shown in Fig. 4(a) sigmoid function's accuracy was overall higher than Relu. As the number of neurons increase the RMSE first decreases until around 70 neurons and then starts increasing again. As expected, the computation time increases exponentially *w.r.t.* neuron count as depicted in Fig. 4(b). Accordingly, we selected to use 70 neurons in LSTM network for sequence

mmeybodi@aut.ac.ir

processing and used the sigmoid activation function that minimizes the RMSE value in the experiments.

Then, we compared the effect of mini-batch size on prediction results. The RMSE values of predictions are evaluated for 3 mini-batch sizes of 90, 180, 360 min over 2, 7, and 17 attributes. As shown in Fig. 5 larger number of attributes improve the prediction results whereas, smaller batch sizes result in lower RMSE values. This can be attributed to the increase in complexity of the system (higher dimensions) without giving the model enough data to match this complexity.

Although the training data is the same for all the mini-batches, the prediction results are different due to the memory of the network. Figure 5 shows trade-offs between batch size and number of features. Although smaller mini-batch sizes may result in smaller RMSE value, larger number of attributes improves the accuracy of prediction by learning the interdependencies better in higher dimensions. This shows the importance of locality in sequential multivariate time-series forecasting problems that can be obtained using networks with memory. The rest of the plot justifies and supports our explanation. In our current scenario, the 17 attributes correspond to all the material flow lines, thus representing a holistic view of the simplified plant model that is learned by the LSTM network.

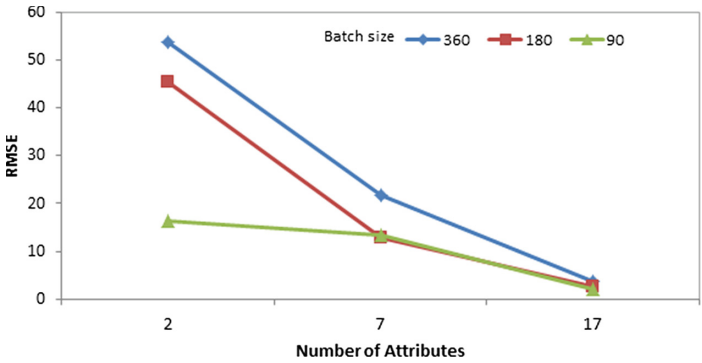


Fig. 5. Effect of mini-batch size and number of attributes on RMSE of predicted values in LSTM network.

5 Conclusions and Future Work

In this paper, we studied the trade-offs between batch size and number of features and their effect on prediction results of multivariate industrial sensor data. We also showed how a time-series dataset can be transformed into a format that is usable in LSTM time-series (*i.e.* deep learning) forecasting. The spatial relation between measurements of time-series data is studied by sequence analysis using

mmeybodi@aut.ac.ir

2 layered LSTM network. The network learns interdependent features from prior raw data to predict future values for industrial supply forecasting. Specifically, we learned the importance of spatio-temporal locality and the need for holistic views for detecting patterns using stacked LSTM networks. In our future work, we will use the LSTM network's predicted values for error detection and classification.

Acknowledgments. This research was sponsored by a grant from TÜPRAŞ (Turkish Petroleum Refineries Inc.) R&D group. We would like to thank Burak Aydoğan and Mehmet Aydin for collecting and providing us with sensor data.

References

1. Lai, G., Chang, W.C., Yang, Y., Liu, H.: Modeling long-and short-term temporal patterns with deep neural networks. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 95–104 (2018)
2. Langkvist, M., Karlsson, L., Loutfi, A.: A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recogn. Lett. **42**, 11–24 (2014)
3. Jiang, Q., Tang, C., Chen, C., Wang, X., Huang, Q.: Stock price forecast based on LSTM neural network. In: International Conference on Management Science and Engineering Management, pp. 393–408. Springer (2018)
4. Varol, G., Laptev, I., Schmid, C.: Long-term temporal convolutions for action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **40**(6), 1510–1517 (2018)
5. Laptev, N., Yosinski, J., Li, L.E., Smyl, S.: Time-series extreme event forecasting with neural networks at Uber. In: International Conference on Machine Learning, vol. 34, pp. 1–5 (2017)
6. Groß, W., Lange, S., Bödecker, J., Blum, M.: Predicting time series with space-time convolutional and recurrent neural networks. In: Proceeding of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, pp. 71–76 (2017)
7. Lee, K.B., Cheon, S., Kim, C.O.: A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. IEEE Trans. Semicond. Manuf. **30**(2), 135–142 (2017)
8. Troiano, L., Villa, E.M., Loia, V.: Replicating a trading strategy by means of LSTM for financial industry applications. IEEE Trans. Ind. Inform. **14**(7), 3226–3234 (2018)
9. Shih, S.Y., Sun, F.K., Lee, H.Y.: Temporal pattern attention for multivariate time series forecasting. arXiv preprint arXiv:1809.04206 (2018)
10. TÜPRAŞ Refinery. http://tupras.com.tr/en/rafineries. Accessed 6 Dec 2018
11. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control. Wiley, Hoboken (2015)
12. Loganathan, G., Samarabandu, J., Wang, X.: Sequence to sequence pattern learning algorithm for real-time anomaly detection in network traffic. In: 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), pp. 1–4 (2018)
13. Khodabakhsh, A., Ari, I., Bakir, M., Ercan, A.O.: Multivariate sensor data analysis for oil refineries and multi-mode identification of system behavior in real-time. IEEE Access **6**, 64389–64405 (2018)

mmeybodi@aut.ac.ir

14. Khodabakhsh, A., Ari, I., Bakir, M., Alagoz, S.M.: Stream analytics and adaptive windows for operational mode identification of time-varying industrial systems. In: 2018 IEEE International Congress on Big Data (BigData Congress), pp. 242–246 (2018)
15. Abadi, M., Barham, P., Chen, J., Chen, Z., et al.: TensorFlow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, pp. 265–283 (2016)
16. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

mmeybodi@aut.ac.ir



Identifying Cancer-Related Signaling Pathways Using Formal Methods

Fatemeh Mansoori[1], Maseud Rahgozar[1(✉)], and Kaveh Kavousi[2]

[1] Database Research Group, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, 11155-4563 Tehran, Iran
{fmansoori, rahgozar}@ut.ac.ir

[2] Complex Biological Systems and Bioinformatics Lab (CBB), Bioinformatics Department, University of Tehran, 1417466191 Tehran, Iran
kkavousi@ut.ac.ir

Abstract. Methods called pathway analysis have emerged whose purpose is to identify significantly impacted signaling pathways in a given condition. Most of these methods employ graphs to model the interactions between genes. Graphs have some limitations in accurately modeling various aspects of the interactions in the signaling pathways. As a result, formal methods as practiced in computer science is suggested for modeling signaling pathways. Using formal methods, various types of interactions among biological components are modeled, which can reduce the false-positive rates compared to other methods. Formal methods can also model the concurrent and stochastic behavior of signaling pathways.

In this article, we illustrate how to employ a formal method for pathway analysis and then to evaluate its performance compared to other methods. Results show that the false-positive rate of a formal method approach is lower than other well-known methods. It is also shown that a formal method approach can identify impacted pathways in pancreatic cancer effectively. Furthermore, it can successfully recognize expecting pathways differentiated between African-American and European-American patients in prostate cancer.

Keywords: Pathway analysis · Enrichment analysis · Formal methods

1 Introduction

Gene expression pattern in control vs. disease samples are routinely used to study disease. This comparison usually results in an extensive list of genes, typically in the order of hundreds or thousands that make it difficult to analyze the effect of each one individually. In this situation, translating the list of genes into biological knowledge is very helpful. For example, cancer is a disease of the genome associated with an aberrant iteration that leads to dysregulation of the cell signaling pathways. It is not clear how genomic changes feed into generic pathways that underlie cancer phenotypes. Therefore, some methods have been developed to summarize the gene expression data into meaningful ranked sets.

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 130–141, 2020.
https://doi.org/10.1007/978-3-030-37309-2_11
mmeybodi@aut.ac.ir

An example is to identify a set of genes that function in the same pathways, which is commonly referred to as pathway analysis. This analysis is useful because it reduces the complexity into pathways level, which is easier to analysis than the gene level. Also, it facilitates identifying signaling pathways relevant to a given disease, which can assist in understanding its mechanisms, develop better drugs production, and personalized drug regimens.

Two types of data are usually used with pathway analysis methods as inputs: the experimental data, like differentially expressed genes obtained when comparing two conditions and the pathway knowledge, that was previously known and stored in pathway annotations databases such as KEGG [1], BioCarta/NCI-PID [2], PANTHER [3] and Reactome [4].

Methods of pathway analysis are divided into three categories [5]. Overexpression analysis (ORA) methods, such as Onto-Express [6] determined impacted pathways according to the number of DEGs. These methods investigate how much the number of differentially expressed genes in a given pathway is significantly higher than those expected randomly.

ORA methods are usually represented by the hypergeometric model or Fisher's exact test. Since these methods require a strict cut-off to determine the differentially expressed genes, their results are strongly affected by the chosen threshold.

Functional Class Scoring (FCS) methods, such as gene set enrichment analysis (GSEA) [7] do not depend on the application of any thresholds. These methods first assign a score to genes and then transform gene scores into pathway scores. ORA and FCS methods treat pathways as lists of genes. However, genes almost do not act independently. Consequently, a new category of methods named topology-based (PT-based) methods have been proposed. Tarca et al. [8] introduced a signaling pathway impact analysis (SPIA) that was the first PT-based method.

PT-based approaches add pathway topology in their analysis for utilizing the correlation between pathway components. Nevertheless, most of the well-known PT-based methods use simple graphs to model the biological pathways [9]. In this type of modeling, Genes and the interactions among them are modeled as nodes and edge respectively, which has some limitations: First, in graph modeling, +1 and −1 weighted edge is used for activation and inhibition relations respectively. This modeling does not accurately reflect properly various situations in which a protein/gene has some activators and inhibitors. When an inhibitor binds to a particular protein, it stops the activation of the protein even in the presence of its activators. Second, in some situations, the simultaneous presence of some proteins/genes together can activate another protein/gene, which is hard to model with a simple graph. Third, if a pathway is triggered through a single receptor and that particular receptor is not expressed, then the pathway will be probably entirely shut off [8]. Fourth, modeling the concurrent and stochastic behavior of signaling pathways is not possible using a graph.

To address the above problems, Mansoori et al. [10] propose a method, named FoPA, using formal methods, as practiced in computer science. This method employs PRISM language for modeling signaling pathways. This approach of modeling signaling pathways has many advantages over those using graphs. It helps to express various relations among biological components involved in an interaction, that leads to

mmeybodi@aut.ac.ir

making a more reliable model of signaling pathways. So, it can be more effective in reducing the false-positive results in pathway analysis studies.

In this article, we outline the general steps required to use formal methods in pathway analysis. Also, we apply this approach to two datasets to illustrates the effectiveness of this modeling approach with formal methods in finding the impacted pathways.

2 Materials and Methods

2.1 Formal Approach

We illustrate a framework, in Fig. 1, to infer significantly impacted pathways in a given clinical condition. In this approach, two lists of genes (e.g., R and R') associated with the desired phenotypes (normal and diseased) and all signaling pathways of KEGG are given as inputs. The problem is to infer pathways that are relevant to differentially expressed genes between R and R'. The result of the approach is a list of pathways sorted from the most relevant to the least relevant.

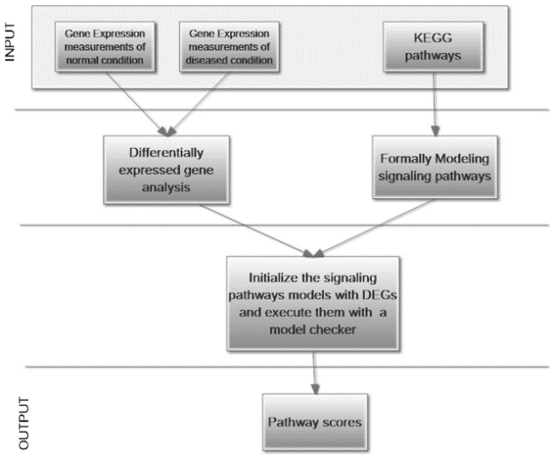


Fig. 1. The framework suggested for formal method approach: The inputs of the formal method approach are two lists of genes associated with the desired phenotypes and the signaling pathways of KEGG. The output is the pathways scores used to rank them according to their relevance to the differential genes. The formal approach requires a formal model of the signaling pathways. The initial configuration of the model is defined using the differentially expressed genes. Once the model is constructed, a model checker is used to execute the model and compute the desired probabilities, that are used to rank pathways.

The formal method approach requires a formal model of the signaling pathways formulated in a formal language. This model defines the evolution of possible configurations of signaling pathways over time. Each configuration of the model is defined using the states of its genes at each time instance. Thus, as the first step, each KEGG signaling pathways are converted into a distinct formal model which can be done with a formal

mmeybodi@aut.ac.ir

language. Any interaction between genes that are important for the analysis should be modeled. Then, an initial state should be defined from which the model checkers start to execute the model. Finally, a score would be assigned to each model based on the result of its execution by the model checker. This score is used to rank pathways according to their relevance to the desired condition. In the following, we explain how to build a simple model for signaling pathways and then how to assign a score to each model.

To represent a pathway using a formal language, different states are defined for each gene. These states would reflect the differential activity of the genes. Suppose, these states are: '*not differentially expressed*', '*not differentially activated*', '*differentially expressed*', and '*differentially activated*'. Then, it should be indicated how the possible states of the system (i.e., the states of all genes) evolve over time. The interactions between genes in signaling pathways change the state of the genes. Suppose, these interactions are activation and inhibition. In activation relations ($A \rightarrow B$), the gene A activates the gene B. If A is an *activated* gene, it can activate the *not activated* gene B, and if A is *differentially activated* or B is *differentially expressed* then B will be *differentially activated*. In inhibition relation ($A \dashv B$), the *activated* gene A prevents the activation of gene B. It means, if gene A and B are *activated*, then gene A leads to the *deactivation* of gene B. if A or B or both of them are *differentially expressed*, then the activated gene A *differentially deactivate* the *activated* gene B.

To make the model probabilistic, we also define a probability for each relation. The Probability, *prob*, for activation relation ($A \rightarrow B$), means that A activates B with probability *prob*, and likewise, it is for inhibition relations.

After constructing the model, the initial state for executing the model should be defined. The initial state of the model is the combination of the initial state of its genes obtained by differentially expression analysis of the disease and normal samples.

To compute a score for each pathway, model checking is used. Model-checking is an automatic verification technique for finite-state concurrent systems that checks whether a model meets specified properties by exploring all possible executions of that. For each signaling pathway model, we employ a model checking tool to compute the probability of differentially activating genes that lead to a cellular response. This is done by describing the appropriate properties of the model in temporal logic. The property should indicate how likely in the future the final effector gene (the gene that leads to a cellular response) will be activated differentially. The probability of activating each of the final effector genes are added to the pathway score.

The pathway score is intended to provide the amount of change incurred by the pathway between two conditions (e.g., normal and diseased). However, this change can take place randomly. Therefore, an assessment of the significance of the measured probability is required. The significance of pathway score is assessed by permuting the label of the normal and disease samples. The distribution of pathway scores from permuted samples is used as a null distribution to estimate the significance of scores as follows:

$$P_F = \frac{\sum_{perm} \mathrm{I}(Score_{perm} \geq Score_{real_sample})}{N_{perm}} \quad (1)$$

mmeybodi@aut.ac.ir

where I(.) is an indicator function, $Score_{perm}$ is the score of the pathway for each permutation, $Score_{real_sample}$ is the score of the pathway for the original data and N_{perm} is the number of permutations.

Different methods can be proposed according to this approach where their differences would be as follows: how to define different states for each gene, how to model the different types of relations between genes and which relations are modeled, how to assign probabilities to each relation and how the property is defined so that by checking that through model checking a score is assigned to each pathway.

The previously mentioned FoPA method [10] is a sample of using formal methods in pathway analysis. In this method, five states are dedicated to each gene, which is *no expression*, *expression*, *differentially expression*, *not differentially activated* and *differentially activated*. The Activation, Inhibition, Phosphorylation activation, Phosphorylation inhibition, Dephosphorylation activation, Dephosphorylation inhibition interactions are modeled with PRISM modeling language. Probability of interactions is computed as a coefficient of the probability of each gene in the probability of the binary relation of genes. The property is defined as the probability that final effector genes are differentially activated eventually in the future.

3 Results and Discussion

Here, we re-examine the FoPA method proposed in [10] with new datasets to evaluate the efficiency of a formal method in finding the impacted pathways.

Among the methods compared in [10], PADOG [11] performs as best as FoPA in some evaluation; therefore, it is chosen for comparison here, too. Moreover, signaling pathway impact analysis (SPIA) [8] is chosen for comparison, because, it is the first introduced PT-based method and almost, all other method are compared with SPIA.

3.1 False-Positive Rate

Because, there is no knowledge of all relevant pathways associated with the conditions, the simulated false inputs are chosen as a set of negative controls. In this experiment, 50 trails are used wherein the class labels (e.g., normal, disease) of the true samples are randomly permuted before the analysis. The percentage mean of the significant pathways (p-value < significant threshold) for permuted samples is expressed as the false positive rate of the method.

Two datasets (GSE8671 [12] and GSE6956 [13]) from Gene Expression Omnibus (GEO) are chosen, and their normal and disease samples are permuted 50 times. For each permuted sample and each of the compared methods, the significant pathways (pathways with a p-value lower than the significant threshold) is counted. The significant mean of these numbers is shown in Table 1 which indicates that the Formal approach false-positive rate is less than that of other methods.

mmeybodi@aut.ac.ir

Table 1. Comparing false-positive rates produced by three methods: The False positive rate for each method and each threshold is obtained by calculating the percentage of the pathways with the p-value below the specified threshold.

Method	Threshold		
	0.01	0.05	0.1
Formal	**0.3**	**2.26**	**5.16**
PADOG	2	6.26	10.84
SPIA	5.95	9.25	13.69

3.2 Pathways Ranking on Real Datasets

In this experiment, we evaluated the ability of the methods to detect potentially relevant signaling pathways. We applied each method to two real data samples. The first one is the Pancreatic ductal adenocarcinoma (PDAC) (GSE32676: mRNA & micro RNA (miRNA) expression in 25 early-stage PDAC [14, 15]), the second one is the prostate cancer dataset (GSE6956: tumor differences in prostate cancer between African-American and European-American men [13, 16]). KEGG pathways are sorted for each dataset in ascending order according to their p-values. For each dataset, we identify pathway(s) which are very likely to be relevant. We want to emphasize the word "likely" because there is no a-priory knowledge of the relevant pathways.

For each of the datasets a matching pathway exists in KEGG, for example for PDAC dataset, the '*pancreatic cancer pathway*' exist in KEGG which is considered as one of the relevant pathways.

PDAC Dataset

This dataset contains mRNA and miRNA expression in 25 early-stage pancreatic ductal adenocarcinomas (PDAC). PDAC is one of the most deadly types of cancers. Early detection of PDAC is very important to improve the prognosis of PDAC.

It is showed that the PI3K pathway activation is critical for the onset and acceleration of PDAC tumors in mice [15]. The other pathway that its activation is required to initiate PDAC is the Wnt signaling pathway. It is showed that this pathway is critical for the progression of pancreatic cancer [17].

PDACs also express high levels of vascular endothelial growth factor (VEGF). Studies indicate that suppression of VEGF expression reduces pancreatic cancer cell tumorigenicity in nude mouse model [18].

Diabetes mellitus is also considered as one of the risk factors for PDAC. A study revealed that new-onset diabetes could potentially indicate early-stage PDAC [19]. Accordingly, the Type II diabetes mellitus pathway is considered related to this disease.

Prostate Cancer Dataset

The prostate cancer dataset contains data for African-American and European-American patients. Evidence indicates that the incidence and mortality rates of prostate cancer in African-American are significantly higher than European-American men. Several research groups suggest that androgen activity is higher in African-American than in Caucasian. In [20], it indicates that AMPK signaling is required for androgen-mediated prostate cancer cell growth and is elevated in prostate cancer. In another

mmeybodi@aut.ac.ir

Formal

Rank	KEGG pathway	pvalue
1	Renal cell carcinoma (05211)	0
2	Acute myeloid leukemia (05221)	0
3	Arrhythmogenic right ventricular cardiomyopathy (05412)	0
4	**PI3K-Akt signaling pathway (04151)**	**0**
5	Prolactin signaling pathway (04917)	0
6	**Pancreatic cancer (05212)**	**0**
7	**Wnt signaling pathway (04310)**	**0**
8	Endometrial cancer (05213)	0
9	**VEGF signaling pathway (04370)**	**0.0097**
10	Vasopressin-regulated water reabsorption (04962)	0.0097
11	T cell receptor signaling pathway (4660)	0.019
12	Neurotrophin signaling pathway (04722)	0.019
13	Tight junction (04530)	0.019
14	Axon guidance (04360)	0.019
15	Intestinal immune network for IgA production (04672)	0.019
⋮	⋮	⋮
26	**Type II diabetes mellitus (04930)**	**0.04**

PADOG

Rank	KEGG pathway	P_{PADOG}
1	Adherens junction (04520)	0.002
2	Pathways in cancer (05200)	0.005
3	**Pancreatic cancer (05212)**	**0.005**
4	Endometrial cancer (05213)	0.008
5	Malaria (05144)	0.01
6	Tight junction (04530)	0.016
7	Acute myeloid leukemia (05221)	0.022
8	ErbB signaling pathway (04012)	0.023
9	Small cell lung cancer (05222)	0.026
10	Bacterial invasion of epithelial cells (05100)	0.03
11	Aldosterone-regulated sodium reabsorption (04960)	0.035
12	Thyroid cancer (05216)	0.038
13	p53 signaling pathway (04115)	0.044
14	Prostate cancer (05215)	0.046
15	**Type II diabetes mellitus (04930)**	**0.056**
⋮	⋮	⋮
38	**VEGF signaling pathway (04370)**	**0.26**
40	**Wnt signaling pathway (04310)**	**0.27**

SPIA

Rank	KEGG pathway	P_{SPIA}
1	Chemokine signaling pathway (04062)	1.06e-07
2	Cytokine-cytokine receptor interaction (04060)	2.78e-06
3	Small cell lung cancer (05222)	1.98e-05
4	NF-kappa B signaling pathway (4064)	3.01e-05
5	Legionellosis (05134)	3.38e-05
6	Osteoclast differentiation (04380)	1.65e-04
7	T cell receptor signaling pathway (04660)	7.7e-04
8	ECM-receptor interaction (04512)	8.64e-04
9	Phosphatidylinositol signaling system (04070)	2.2e-03
10	Pathways in cancer (05200)	2.94e-03
11	**Pancreatic cancer (05212)**	**5.64e-03**
12	Amoebiasis (05146)	6.92e-03
13	**Type II diabetes mellitus (04930)**	**8.3e-03**
14	p53 signaling pathway (04115)	1.17e-02
15	Transcriptional misregulation in cancer (05202)	1.2e-02
⋮	⋮	⋮
36	**VEGF signaling pathway (04370)**	**0.057**
39	**Wnt signaling pathway (04310)**	**0.08**

Fig. 2. The top 15 pathways retrieved by the Formal approach, PADOG and SPIA for PDAC (GSE32676) dataset. '*PI3K-ACT signaling pathway*', '*VEGF signaling pathway*', '*Wnt signaling pathway*' '*Type II diabetes mellitus*' and '*Pancreatic cancer*' pathways which are shown in bold are expected to be impacted by PDAC.

mmeybodi@aut.ac.ir

Formal

Rank	KEGG pathway	pvalue
1	Serotonergic synapse (04726)	0
2	**Prolactin signaling pathway (04917)**	**0**
3	Neurotrophin signaling pathway (04722)	0
4	Adherens junction (0520)	0
5	Melanogenesis (04916)	0
6	Renin secretion (04924)	0
7	**Prostate cancer (05215)**	**0.005**
8	p53 signaling pathway (04115)	0.0097
9	Long-term depression (04730)	0.0097
10	**AMPK signaling pathway (04152)**	**0.01**
11	Acute myeloid leukemia (05221)	0.011
12	Colorectal cancer (05210)	0.015
13	Small cell lung cancer (05222)	0.019
14	Insulin signaling pathway (04910)	0.019
15	**Estrogen signaling pathway (04915)**	**0.019**

PADOG

Rank	KEGG pathway	P_{PADOG}
1	SNARE interactions in vesicular transport (04130)	0.0003
2	RNA degradation (03018)	0.0003
3	Neurotrophin signaling pathway (04722)	0.0003
4	Pancreatic cancer (05212)	0.0003
5	Long-term potentiation (04720)	0.0003
6	Vibrio cholerae infection (05110)	0.032
7	Epithelial cell signaling in Helicobacter pylori infection (05120)	0.032
8	Gap junction (04540)	0.032
9	GnRH signaling pathway (04912)	0.032
10	VEGF signaling pathway (04370)	0.032
11	Calcium signaling pathway (04020)	0.064
12	Adherens junction (04520)	0.064
13	Insulin signaling pathway (04910)	0.064
14	Protein processing in endoplasmic reticulum (04141)	0.097
15	Vasopressin-regulated water reabsorption (04962)	0.097
⋮	⋮	⋮
19	**Prostate cancer (05215)**	**0.097**

SPIA

Rank	KEGG pathway	pvalue
1	GnRH signaling pathway (04912)	0.0002
2	Protein processing in endoplasmic reticulum (04141)	0.015
3	Bacterial invasion of epithelial cells (05100)	0.019
4	Prion diseases (05020)	0.023
5	Melanoma (05218)	0.025
6	Endocrine and other factor-regulated calcium reabsorption (04961)	0.041
7	Antigen processing and presentation (04612)	0.049
8	Epithelial cell signaling in Helicobacter pylori infection (05120)	0.05
9	Thyroid cancer (05216)	0.073
10	Renal cell carcinoma (05211)	0.077
11	Chagas disease (American trypanosomiasis) (05142)	0.077
12	Pathways in cancer (05200)	0.078
13	Endometrial cancer (05213)	0.083
14	Bladder cancer (05219)	0.093
15	Leishmaniasis (05140)	0.097
	…	…
25	**Prostate cancer (05215)**	**0.2**

Fig. 3. The top 15 pathways retrieved by the formal approach, PADOG and SPIA for the prostate cancer (GSE6956) in African-American. '*AMPK signaling pathway*', '*Estrogen signaling pathway*', '*prolactin signaling pathway*' and '*prostate cancer*' shown in bold are expected to be impacted in these samples.

mmeybodi@aut.ac.ir

Formal

Rank	KEGG pathway	pvalue
1	p53 signaling pathway (04115)	0
2	Wnt signaling pathway (04310)	0
3	Apoptosis (04210)	0
4	Neurotrophin signaling pathway (04722)	0
5	Adherens junction (04520)	0
6	Regulation of lipolysis in adipocytes (04923)	0.009
7	Circadian rhythm (04710)	0.009
8	Amyotrophic lateral sclerosis (ALS) (05014)	0.019
9	Inflammatory bowel disease (05321)	0.029
10	Retrograde endocannabinoid signalling (04723)	0.029
11	**Prolactin signaling pathway (04917)**	**0.038**
12	TNF signaling pathway (04668)	0.038
13	Non-alcoholic fatty liver disease (NAFLD) (04932)	0.038
14	ErbB signaling pathway (04012)	0.045
15	**Prostate cancer (05215)**	**0.045**

PADOG

Rank	KEGG pathway	P_{PADOG}
1	Renal cell carcinoma (05211)	3.9e-05
2	Adherens junctionn (04520)	1.18e-02
3	Chronic myeloid leukemia (05220)	1.18e-02
4	GnRH signaling pathway (04912)	2.7e-02
5	Notch signaling pathway (04330)	3.14e-02
6	ErbB signaling pathway (04012)	2.53e-02
7	Acute myeloid leukemia (05221)	3.92e-02
8	Type II diabetes mellitus (04930)	5.09e-02
9	**Prostate cancer (05215)**	**5.49e-02**
10	Non-small cell lung cancer (05223)	5.88e-02
11	Adipocytokine signaling pathway (04920)	6.27e-02
12	Insulin signaling pathway (04910)	6.66e-02
13	Fc gamma R-mediated phagocytosis (04666)	6.66e-02
14	Neurotrophin signaling pathway (04722)	7.45e-02
15	Thyroid cancer (05216)	8.23e-02

SPIA

Rank	KEGG pathway	pvalue
1	GnRH signaling pathway (04912)	4.34e-06
2	HTLV-I infection (05166)	5.3e-05
3	TGF-beta signaling pathway (04350)	1.92e-04
4	Small cell lung cancer (05222)	2.13e-04
5	Cell cycle (04110)	3.34e-04
6	Endocrine and other factor-regulated calcium reabsorption (04961)	4.3e-04
7	Fc gamma R-mediated phagocytosis (04666)	5.4e-04
8	Renal cell carcinoma (05211)	6.57e-04
9	Basal cell carcinoma (05217)	7.67e-04
10	Chronic myeloid leukemia (05220)	8.4e-04
11	T cell receptor signaling pathway (04660)	8.69e-04
12	B cell receptor signaling pathway (04662)	0.001
13	Fc epsilon RI signaling pathway (04664)	0.0015
14	Bacterial invasion of epithelial cells (05100)	0.0016
15	Type II diabetes mellitus (04930)	0.0022
	…	…
38	**Prostate cancer (05215)**	**0.052**

Fig. 4. The top 15 pathways retrieved by the formal approach, PADOG, and SPIA for the prostate cancer in European-American (GSE6956) dataset. '*Prolactin signaling pathway*' and '*prostate cancer*' pathway shown in bold are expected to be impacted in these samples.

mmeybodi@aut.ac.ir

analysis, it indicates that African–American men had significantly higher serum estradiol levels than Caucasian or Mexican–American men [21]. Therefore, we expected that the '*AMPK signaling pathway*' and '*Estrogen signaling pathway*' to be up-regulated in African-Americans prostate cancer.

Since, there is the body of evidence that strongly supports the contribution of the prolactin receptor (PRLR) signaling in breast and prostate tumorigenesis and cancer progression [22, 23], the '*prolactin signaling pathway*' is likely to be relevant to both African-American and European-Americans prostate cancer.

Three pathway analysis methods are compared regarding their ability to identify the expected relevant pathways for two datasets. The 15 top relevant pathways identified by Formal, PADOG and SPIA methods for PDAC dataset are shown in Fig. 2. As illustrated, the formal approach identifies the *'PI3K-ACT' signaling pathway*, '*Wnt signaling pathway*' and '*VEGF signaling pathway*', '*Pancreatic cancer*' and '*Type II diabetes mellitus*' pathway with a lower rank than the other compared methods.

Figure 3 shows 15 top identified pathways for African-American samples of prostate cancer dataset. The '*AMPK signaling pathway*', '*Estrogen signaling pathway*', '*prolactin signaling pathway*' and '*prostate cancer*' pathway are among the 15 top relevant pathways identified by the formal approach.

Figure 4 shows 15 top identified pathways for European-American samples of prostate cancer dataset. The formal approach identified the '*prolactin signaling pathway*' and '*prostate cancer*' as more relevant to these samples of prostate cancer. As it is clear, the '*AMPK signaling pathway*' and '*Estrogen signaling pathway*' are not identified for European-American patients.

4 Conclusion

In this study, we presented how to use formal methods for pathway analysis. Despite the other methods that used simple graphs for modeling signaling pathways, we use formal methods. Formal modeling has multiple advantages compared to the methods using graphs. It helps researchers to express various types of relations among the biological components involved in the same interaction. This helps to create a more realistic model of signaling pathways, which can also reduce the false-positive rates of the pathway analysis method. We compare a sample of our approach for pathway analysis with two topology-based (PADOG, SPIA) analysis methods.

The simulated false inputs (permuted class labels) are created as a set of negative controls to test the false-positive rate of the methods. The number of significant pathways identified by giving permuted class labels to the formal approach is less than the other two methods; that is, the formal approach can discriminate better between actual and random input data. For further evaluating the proposed approach, we applied it to two real datasets (pancreatic cancer and prostate cancer datasets). We showed that our approach discovered pathways expected to be relevant to these datasets effectively.

These lines of evidence, well demonstrated the advantage of the proposed approach over other methods. The only disadvantage of formal approach may be its high running time compared with statistical methods. While the running time is not a concern in pathway analysis methods, this is not a case that bothers researchers.

mmeybodi@aut.ac.ir

References

1. Kanehisa, M., Goto, S.: KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. **28**(1), 27–30 (2000)
2. Schaefer, C.F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., Buetow, K.H.: PID: the pathway interaction database. Nucleic Acids Res. **37**(suppl_1), D674–D679 (2008)
3. Mi, H., Lazareva-Ulitsky, B., Loo, R., Kejariwal, A., Vandergriff, J., Rabkin, S., Kitano, H.: The PANTHER database of protein families, subfamilies, functions, and pathways. Nucleic Acids Res. **33**(suppl_1), D284–D288 (2005)
4. Croft, D., O'Kelly, G., Wu, G., Haw, R., Gillespie, M., Matthews, L., Jupe, S.: Reactome: a database of reactions, pathways and biological processes. Nucleic Acids Res. **39**(suppl_1), D691–D697 (2010)
5. Khatri, P., Sirota, M., Butte, A.J.: Ten years of pathway analysis: current approaches and outstanding challenges. PLoS Comput. Biol. **8**(2), e1002375 (2012)
6. Draghici, S., Khatri, P., Tarca, A.L., Amin, K., Done, A., Voichita, C., Romero, R.: A systems biology approach for pathway level analysis. Genome Res. **17**(10), 1537–1545 (2007)
7. Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Mesirov, J.P.: Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. **102**(43), 15545–15550 (2005)
8. Tarca, A.L., Draghici, S., Khatri, P., Hassan, S.S., Mittal, P., Kim, J.S., Romero, R.: A novel signaling pathway impact analysis. Bioinformatics **25**(1), 75–82 (2008)
9. Mitrea, C., Taghavi, Z., Bokanizad, B., Hanoudi, S., Tagett, R., Donato, M., Draghici, S.: Methods and approaches in the topology-based analysis of biological pathways. Front. Physiol. **4**, 278 (2013)
10. Alur, R., Henzinger, T.A.: Reactive modules. Formal Methods Syst. Des. **15**(1), 7–48 (1999)
11. Tarca, A.L., Draghici, S., Bhatti, G., Romero, R.: Down-weighting overlapping genes improves gene set analysis. BMC Bioinform. **13**(1), 136 (2012)
12. GEO Accession Viewer. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE8671. Accessed 7 Dec 2018
13. GEO Accession Viewer. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE6956. Accessed 7 Dec 2018
14. GEO Accession Viewer. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE32676. Accessed 7 Dec 2018
15. Donahue, T.R., Tran, L.M., Hill, R., Li, Y., Kovochich, A., Calvopina, J.H., Li, X.: Integrative survival-based molecular profiling of human pancreatic cancer. Clin. Cancer Res. **18**(5), 1352–1363 (2012)
16. Wallace, T.A., Prueitt, R.L., Yi, M., Howe, T.M., Gillespie, J.W., Yfantis, H.G., Ambs, S.: Tumor immunobiological differences in prostate cancer between African-American and European-American men. Can. Res. **68**(3), 927–936 (2008)
17. Zhang, Y., Morris, J.P., Yan, W., Schofield, H.K., Gurney, A., Simeone, D.M., di Magliano, M.P.: Canonical Wnt signaling is required for pancreatic carcinogenesis. Can. Res. **73**(15), 4909–4922 (2013)
18. Korc, M.: Pathways for aberrant angiogenesis in pancreatic cancer. Mol. Cancer **2**(1), 8 (2003)
19. Kanno, A., Masamune, A., Hanada, K., Kikuyama, M., Kitano, M.: Advances in early detection of pancreatic cancer. Diagnostics **9**(1), 18 (2019)

mmeybodi@aut.ac.ir

20. Tennakoon, J.B., Shi, Y., Han, J.J., Tsouko, E., White, M.A., Burns, A.R., Zhang, A., Xia, X., Ilkayeva, O.R., Xin, L., Ittmann, M.M.: Androgens regulate prostate cancer cell growth via an AMPK-PGC-1α-mediated metabolic switch. Oncogene **33**(45), 5251 (2014)
21. Rohrmann, S., Nelson, W.G., Rifai, N., Brown, T.R., Dobs, A., Kanarek, N., Platz, E.A.: Serum estrogen, but not testosterone, levels differ between black and white men in a nationally representative sample of Americans. J. Clin. Endocrinol. Metab. **92**(7), 2519–2525 (2007)
22. Goffin, V.: Prolactin receptor targeting in breast and prostate cancers: new insights into an old challenge. Pharmacol. Ther. **179**, 111–126 (2017)
23. Hernandez, M.E., Wilson, M.J.: The role of prolactin in the evolution of prostate cancer. Open J. Urol. **2**(03), 188 (2012)

mmeybodi@aut.ac.ir



Predicting Liver Transplantation Outcomes Through Data Analytics

Bahareh Kargar, Vahid Gheshlaghi Gazerani, and Mir Saman Pishvaee(✉)

School of Industrial Engineering,
Iran University of Science and Technology, Tehran, Iran
pishvaee@iust.ac.ir

Abstract. Computer-based learning methods in medical contexts have attracted a great deal of attention recently. Organ transplantation is one of the key areas where prognosis models are being used for predicting the patients' survival. The only treatment for patients who suffer from liver failure is transplantation. The aim of the present study is to model the patients' survival prediction as well as to recognize the most significant attributes on survival after liver transplantation. To address the issue of the imbalanced dataset, a combination of two techniques has been considered to evaluate the result; under-sampling and over-sampling techniques. Decision Tree (DT) and K Nearest Neighbor (KNN) models together with Artificial Neural Network (ANN) have been utilized on the dataset separately to define two-year mortality of patients after liver transplantation using the dataset of Iran Ministry of Health and Medical Education (MOHME). By using Genetic Algorithm (GA), it has been shown that 13 attributes have a strong impact on survival prediction in the case of liver transplant recipients. We also compared three classification models using Receiver Operating Characteristic (ROC) curve and other various performance measures. Moreover, findings of the proposed method have improved the results of previous predictions; Using Decision Tree method, roughly in 80% of the transplantation outcomes have been predicted correctly.

Keywords: Liver transplantation · Survival prediction · Medical data mining · Healthcare analytics

1 Introduction

Liver transplantation (LT) is an appropriate real life-saving treatment for patients with End-stage Liver Disease (ESLD). This treatment which has progressed well over the past 50 years increases the life quality and decreases the death risk in the final stage of liver failure [1]. Survival prediction is a key parameter to identify the success of liver transplantation surgery. The ever-growing gap between supply of and demand for organs leads to the death of some waiting list patients requiring organ transplantation urgently. Currently, candidates are prioritized on the waiting list of cadaveric donor liver transplant medical urgencies. Medical specialists make decisions regarding the liver transplantation, as they predict the transplantation outcomes based on the Model

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 142–160, 2020.
https://doi.org/10.1007/978-3-030-37309-2_12

mmeybodi@aut.ac.ir

for End-stage Liver Disease (MELD) score. The MELD score, which is a fundamental function of the bilirubin, creatinine, and INR of the normalized international ratio, is a short-term prediction model for patients suffering from liver cirrhosis. Using the MELD score, patients who have the highest score in the waiting list, will have the highest preference in liver allocation system [2]. However, some patients receive donor liver immediately, and some must wait a long time for donor organs which results in less chance of survival rate [3]. Since the current liver allocation procedure does not consider any criterion to measure the post-transplant outcome, some liver recipients will not receive a liver that continues to work for them as long as it is needed. Efficiency may decrease in a medical urgency-based method as waiting patients with the extreme pre-transplant death risk may also have the least life expectancy after transplanting. Another weakness of using the MELD score is that some patients with urgent need are neglected, which is called MELD exception. Continuous examination of more precise models to predict the long-term survival of patients who are experiencing liver transplantation led to the introduction of more exact models with high prediction accuracy. New trends in biomedicine are employing data mining as a beneficial tool for a majority of problems, in the last decades, which results in notable applications for science [4, 5].

A massive dataset of patients and donors was gathered in a database, while to predict the survival of recipients, only a very small amount of data was used. In a study of Doyle et al. in which 149 patients who were adults underwent LT at Presbyterian University Hospital, Pittsburgh [6], researchers derived a fact to determine the probability of graft failure in liver patient by using the analysis of stepwise logistic regression. When the authors faced the challenge of introducing a model which explains the nonlinearity among variables, they tried using of a neural network model which was 10 feed-forward back propagation to predict the survival, later in [7]. In 2006, a study was conducted by Cuccheti et al. [3] on 251 consecutive people with cirrhosis referred to LT at one of liver transplantation units in Italy. They demonstrated that ANN was preferable than the MELD score [3]. In [8], Marsh et al. introduced an analysis of survival and time to reappearance of Hepato Cellular Carcinoma (HCC) following Orthotropic LT (OLT) on 214 patients at Pittsburgh Medical Center. They applied a 3-layer feedforward neural network model and concluded that male patients have a higher risk of HCC recurrence than females.

Khosravi et al. [9], utilized neural networks and Cox Proportional Hazard (Cox PH) predicting 5-year survival of patients as well as estimating post-transplantation efficacy features. Their results revealed that neural networks results are more accurate (with an accuracy of 92.73%). In the latest research, Raji et al. [10] used a multi-layer perceptron artificial neural network model to predict patients' survival after liver transplantation in a period of 12 years, using a large dataset of United Network for Organ Sharing (UNOS). The obtained model has an accuracy of 99.74%, which is the highest compared to the previous models. Finally, in [11, 12], a rule-based system was proposed using clinical data from various Spanish liver transplantation units to determine graft survival a year after liver transplant. One of the main restrictions of the proposed methods in [11] is the specific fitness functions which are applied to tune the neural network weights and structure through using the multi-objective evolutionary

mmeybodi@aut.ac.ir

algorithms to deal with the imbalanced dataset. As a result, the corresponding computational cost would be very high.

As indicated in previous research, different donor attributes exist which lead to graft losses or an higher risk [13]. Since there are numerous risk factors which can lead to graft loss, these characteristics and risks should be carefully taken into consideration in the decision support system [14, 15]. Therefore, the aim of this work was to introduce a model to predict liver transplantation survival using data mining algorithms and identifying more influential attributes in the survival of transplanted liver patients using genetic algorithm.

Although the performance of data mining techniques to predict the survival of patients after liver transplantation has been assessed, the imbalanced essence of the data is a restriction still, since the outcomes incline to be worse for the minority one. In fact, class imbalance is one of the most prevalent issues in medical applications [16, 17], in cases that 1 or more than one classes have a far lower chance to be included in the training set. In this research, graft loss is the less frequent class, the main aim is to predict a failure correctly though. This issue must be considered precisely in the model construction phase; otherwise, trivial models (i.e. always the majority class is predicted) may be achieved. Utilizing a re-sampling strategy (under sampling the majority class or oversampling the minority one), this issue will be addressed generally. In this current study, combining these two common approaches including an under-sampling technique and an over-sampling technique is recommended, which may contribute to classification performance improvement on imbalanced datasets.

This paper is structured as the following: Sect. 2 covers the presented methodology, explaining data pre-processing stage, as well as the technique used for selecting attributes. A simulation of the proposed classification models is presented in Sect. 3 and following by that in Sect. 4, the experimental results have been presented. Finally, there are future research directions and conclusion in Sect. 5.

2 Materials and Methods

In recent decades, there has been significant improvements in terms of the quantity and quality of transplantation types, in Iran. By definition, the patients' survival after LT means the patient receives the maximum benefit of the organ transplantation, so they can be most likely to live longer with a successful organ transplantation. Therefore, considering several attributes that are influencing this process, including the effectiveness and correctness of the patterns associated with donors and recipients, as well as transplantation surgery itself, may need a rigorous clinical planning which will lead to increasing the chance of survival after the transplantation. Here in this study, the most significant attributes regarding the patient's survival will be determined using Genetic Algorithm.

As mentioned earlier, pervious research has been proved the good performance of utilizing data mining techniques for the patient's survival, a problem of imbalanced dataset still exists. To address this issue, in this study, a pre-processing stage is considered to overcome the imbalanced data problem by using both over-sampling and under-sampling techniques. Subsequently, to evaluate the highest probability of the

mmeybodi@aut.ac.ir

patient's survival, three classification models are applied including Artificial Neural Network, K Nearest Neighbor and Decision Tree. At first place, to build the classifiers, Rapid Miner Studio professional 7.1 and Weka 3.6.9 software have been used, and following by that obtained performance have been compared with several evaluation measures. The obtained outcome has been shown utilizing ROC curves. In Fig. 1, the overall procedure of the presented method for patient's survival prediction is shown after LT. In the following, the dataset attributes as well as all the pre-processing stages will be described.

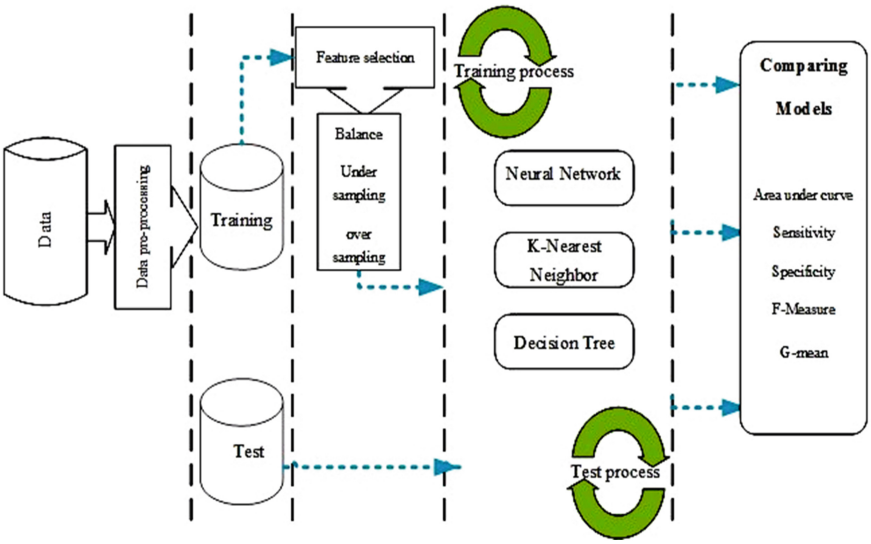


Fig. 1. The overall procedure of the proposed method.

2.1 Dataset Description

Several medical and clinical parameters are involved in the patient's survival. In the current study, by using the Genetic Algorithm only the attributes that will affect the survival of the patient after the transplantation, have been studied. As this research is a descriptive- analytical study, all the data, information and statistics needed for the presented experimental studies have been provided by the Ministry of Health and Medical Education (MOHME) of Iran, Department of Transplantation and Special Diseases. All the data and attributes which are provided in this section have been collected through a census based on experts' opinion regarding the liver transplant cases between the period of two years, from 2011 to 2012. This dataset contains 632 high-risk patients of both genders who underwent a liver transplantation. Some cases are excluded from this research, including people who are undergoing transplantation more than one time, missing some essential data, survived less than one day, or any types of transplantation rejection. In the current study, prediction of patients' survival who underwent liver transplantation and survived obtained about 91.6% (roughly 578

mmeybodi@aut.ac.ir

cases) and around 8.4%, in the case of people who died within two years after trans-plantation (roughly 53 cases).

In total, 38 input attributes and one STATUS output node have been recognized as binary variables. The patient graft status was defined as STATUS = 1 for graft failure and STATUS = 0 for the successful result. The recipient, donor, and trans-plantation attributes have been considered as input for the classification models. Thirty-eight attributes have been considered as input attributes (Independent variables) for each patient, including recipient's age, recipient's weight, comorbidity disease, pack cell (PC), duration of hospital stay, exploration after transplantation, lung complication after transplantation, diabetes after transplantation, Cytomegalovirus (CMV) infection, and post-transplantation vascular complication. An explanation of the qualitative and quantitative attributes is given in Table 1.

2.2 Data Pre-processing

After data collection stage, pre-processing techniques will be applied on data, in the aim of excluding redundant information due to incompleteness or incorrectness. As mentioned earlier, in addition to other phases, the imbalanced nature of the dataset should be considered for preprocessing stage as well. If a classification model employs an imbalanced dataset, it might overlook or at some cases ignore the minority class. To address this issue, two techniques including over-sampling and under-sampling techniques are utilized in the preprocessing step [18, 19]. To clarify, over-sampling is basically a procedure which generates new samples providing an imbalanced dataset [18]. As one of the techniques of producing samples, it simply reproduces the minority class n number of times, so no major or minor class exists. To balance the data, the number of minority examples is increased by over-sampling them. Under-sampling is another sampling data technique that declines the data instances. Selecting a suitable subset of majority class samples randomly is one simple method for under-sampling data [20, 21]. Avoiding the bias towards majority class instances and achieving a high performance of classification are the main objectives of balancing data using under-sampling technique [22]. By using an under-sampling technique, the samples from the majority class ('Alive' class) further will be selected in this study.

Initially, to balance the data, training sets have been separated from test sets. Analyzing the data have been performed by Rapid Miner software, using a random sampling with the ratio of 0.7 and 0.3 for training and test sets, respectively. Subsequently, over sampling and under-sampling techniques have been applied jointly by using Weka software to balance the majority classes versus minority classes and vice versa. Most data mining and classification algorithms have a stronger orientation toward the majority class data, while in medical applications considering the importance of the topic, the aim is to minimize the errors as much as possible, especially in the minority class.

mmeybodi@aut.ac.ir

Table 1. Description composite variables of input attributes.

Input attributes	Type of attributes	Composite variables	Input attributes	Type of attributes	Composite variables
Recipient sex	Nominal	Recipient	**Waiting list time (day)**	Numeric	
Male			**Duration of hospital stay (day)**	Numeric	
Female			**Previous abdominal surgery**	Nominal	
Recipient age (year)	Numeric		No		
Weight (kg)	Numeric		Yes		
Recipient diagnosis disease	Nominal		**Renal failure before transplantation**	Nominal	
Metabolic			No		
Cholestatic			Yes		
Hepatitis			**Diabetes after transplantation**	Nominal	
Tumors			No		
Other causes			Yes		
Cryptogenic			**Vascular complication after transplantation**[a]	Nominal	
Comorbidity disease[b]	Nominal		No		
No			Yes		
Yes			**Renal failure after transplantation**	Nominal	
MELD/PELD score	Nominal		No		
<20			Yes		
≥ 20			**PNF**	Nominal	
Child score	Numeric		No		
Child class	Nominal		Yes		
A			**PTLD**	Nominal	
B			No		
C			Yes		
Creatinine (mg/dl)	Numeric		**CMV infection after transplantation**	Nominal	

(*continued*)

mmeybodi@aut.ac.ir

Table 1. (*continued*)

Input attributes	Type of attributes	Composite variables	Input attributes	Type of attributes	Composite variables
No			**Lung complication after transplantation**[c]	Nominal	
Yes			No		
Pack cell (bag)	Numeric		Yes		
Fresh frozen plasma (bag)	Numeric		**Donor age (year)**	Numeric	Donor
Total bleeding (ml)	Numeric		**Donor sex**	Nominal	
Bile duct complication after transplantation	Nominal		male		
No			Female		
Yes			**Donor**	Nominal	
Exploration after transplantation	Nominal		living		
No			cadaver		
Yes			**Cold ischemia time (hour)**	Numeric	
Acute rejection	Nominal		**Warm ischemia time (hour)**	Numeric	
No			**Donor cause of death**	Nominal	
Yes			Living		
Chronic rejection	Nominal		Trauma		
No			CVA		
Yes			Other		
R-HCC, R-HBV, R-HCV	Nominal		**Type of transplantation**	Nominal	Transplantation
No			Whole organ		
Yes			Split		
Total bilirubin (mg/dl)	Numeric		Partial		
INR (IU)	Numeric		**Duration of Operation(hour)**	Numeric	

[a]Hepatic Artery Thrombosis, Portal Vein Thrombosis, Hepatic Artery Stenosis
[b]Diabetes, Heart Failure and Lung Disease
[c]Diabetes, Heart Failure and Lung Disease
Abbreviations: **PELD**: Pediatric End-Stage Liver Disease, **MELD**: Model for End-Stage Liver Disease, **CHILD Score-Class**: Child-Turcotte-Pugh Score-Class, **INR**: International Normalized Ratio, **PNF**: Primary non-function, **CMV**: Cytomegalovirus, **PTLD**: Post-transplant lymphoproliferative disorder, **R-HCC**: Recurrence of hepatocellular carcinoma, **R-HBV**: Reinfection of Hepatitis B virus, **R-HCV**: Reinfection of Hepatitis C virus.

mmeybodi@aut.ac.ir

2.3 Feature Selection

One of the most critical challenges to create the classification models is to select the appropriate attributes. In practice, real-world data often involve numerous attributes, which some of them may be insignificant for the purpose of classification. Furthermore, in some cases, when creating a new model, a feature selection algorithm adds to the original model to select the most suitable features. Increasing the number of attributes will make the calculation complex. On the other hand, many attributes have correlation that leads to receiving repetitive and redundant information [23]. By applying feature selection technique, significant features are identified for creating models as well having a better interpretation of data. GA is a stochastic general search technique, which is capable of efficiently searching big search areas, which happens usually the case in feature selection procedure.

3 Experimental Study

In the current section, the classification models are initially demonstrated with the aim of predicting post liver transplantation survival, and the evaluated measures. The experimental design has been applied on two datasets including training sets and test sets. Almost 70% of patterns from training set and 30% of patterns from test set have been preserved. Roughly 631 instances and 442 records have been allocated as the training set and 189 records have been reserved as the test set. Prior applying classification models, appropriate attributes are selected by using feature selection methods. By using expert judgment and evaluation by classification models, it has been shown that GA results better for identifying the attributes which are affecting post-transplant survival. The clinical input attributes have been selected by using genetic algorithm in Weka software, which is suitable for classifying and visualizing the datasets [24].

With the help of GA, 13 attributes marked as the important ones among all 38 attributes, following by that these optimal attributes are considered as inputs of the predicted models. The most significant attributes associated with better post-transplantation outcomes are presented in Table 2. Mean/standard deviation is utilized for numerical attributes and the number and percentage for nominal attributes as the attributes representation. For instance, for the recipient's age, in n = 631 model, the probable mean is 33.308 with a standard deviation of 19.34. Following by that, these selected clinical data have been categorized in three classification models as the inputs. Finally, the obtained results are compared according to several evaluation measures.

As it is observed, the recipient's age, PTLD, acute rejection, primary non-function (PNF), renal failure, exploration and lung complication after transplantation are significant attributes in the patients' survival. In addition, some attributes such as previous abdominal surgery, cold ischemia time (CIT), bleeding, INR, total bilirubin and duration of operation are also among the influential attributes.

mmeybodi@aut.ac.ir

Table 2. Important attribute based on GA.

Input attributes	Values
Recipient age (year)	33.3078 ±19.34
INR (IU)	1.98 ± 1.097
Total Bilirubin (mg/dl)	8.045 ± 10.690
Cold ischemia Time (hour)	6.414 ± 3.417
Total Bleeding (ml)	1873.129 ± 2124.615
Duration of Operation (hour)	5.905 ± 1.118
Previous abdominal surgery	
No	554(87.80%)
Yes	77(12.20%)
Renal failure after transplantation	
No	607(96.20%)
Yes	24(3.80%)
PNF	
No	626(99.20%)
Yes	5(.80%)
PTLD	
No	624(98.89%)
Yes	7(1.10%)
Lung Complication after transplantation	
No	614(97.30%)
Yes	17(2.70%)
Exploration after transplantation	
No	509(80.66%)
Yes	122(19.33%)
Acute Rejection	
No	435(68.94%)
Yes	196(31.06%)

3.1 Classification Models in Survival Prediction

In this section, first three different classification models for the use of predicting patients' survival who underwent liver transplantation will be explained and then the achieved result by every classification methodology will be compared with others. In this study, the clinical data of donors, recipients and transplant attributes have been considered as the input to these models. The STATUS is the final output of models, which displays the status of the liver graft as Yes or No. No shows the graft failure and Yes indications the graft survival which is provided in Table 3.

Table 3. Representation of Output Class, STATUS.

STATUS = 0	Yes	Best survival
STATUS = 1	No	Poor survival

mmeybodi@aut.ac.ir

- **Artificial Neural Network:** previous research showed that a Multilayer Perceptron (MLP) model is an influential computational model processing the nonlinear functions and producing the high accuracy outcomes [25]. An MLP includes multiple layers of nodes in a directed graph, with every layer fully-connected to the next layer. Except for the input node, every node is a processing element or neuron with a non-linear activation function. MLP uses back propagation to train the network. This class of networks includes multiple layers of computational units, inter-connected in a feed forward way. The 3 layers including output layer, input layer and hidden layer form the MLP architecture. In the present study, a multilayer perceptron (MLP) was applied consisting of 39 nodes including 38 independent and 1 dependent node as STATUS result node. The model trained the clinical data involving the chosen features of donor, transplantation and recipient based on GA utilizing the back propagation technique. Sigmoid function was the activation function utilized in the hidden layers. The MLP model was trained at a rate of 0.9 and the number of hidden layers was 3 as well as a momentum of 0.1 was applied through training. The number of training cycles used to train the model was 50.
- **Decision Tree**: this methodology has become popular in medical applications recently, due to the virtuous performance and interpretability. The basis of DT design in this paper was the standard C4.5 algorithm [26]. Based on the split criterion, during the training procedure, the dataset has been separated in a recursive manner till the optimal DT hierarchy of nodes obtain. The gained ratio is the split optimization criterion utilized in the proposed DT model. It is utilized to decrease a bias toward multi valued features by considering the size and number of branches when selecting a feature. The minimal gain and maximum depth were set at 0.1 and 1000. The algorithm was used to learn a rule set as opposed to a tree, as this provided slightly higher generalization accuracy.
- **K Nearest Neighbor:** K Nearest Neighbor algorithm is non-parametric decision-making technique used for classification, moreover, it was regarded as one of the most operative algorithms in data mining [27]. K Nearest Neighbor algorithm performs the classification based on the similarity of each record to neighboring records using the Euclidean distance criterion. First of all, initial inputs become normal, using Z-score standardization:

$$x^* = \frac{x - x_{mean}}{SD(x)} \tag{1}$$

This statistical normalization aimed to convert a data into Normal distribution with mean = 0 and variance = 1. If the training sets is an X set and $x_i \in X$, then the value of each record is calculated as:

$$\text{vote} = \frac{1}{d(x_{new}, x_i)^2} \tag{2}$$

mmeybodi@aut.ac.ir

3.2 Performance Metrics

Although cross-validation is a standard procedure for performance evaluation, its joint application with re-sampling strategy (under-sampling and over-sampling techniques) remains an open question for researchers farther from the imbalanced data topic. In this regard, there are different cross-validation approaches: CV after re-sampling and CV during re-sampling. When the cross-validation is implemented after re-sampling is applied, similar patterns may appear in both training and test partitions, leading to overoptimistic error estimates. When the cross-validation is applied during re-sampling, only the training patterns are considered both for generating new patterns and training the model, avoiding overoptimism [28]. In both approaches, similar or exact copies may appear in the training partitions, leading to overfitting. Due to the afore-mentioned reasons explained, a number of other appropriate performance measures can be taken into account for evaluation of classifiers. Accuracy (Acc) or correct classification rate is the most popular measure in data mining. Although, for some situations (e.g. in imbalanced dataset) this metric is not the best choice. Consequently, the accuracy (Acc) is supplemented with G-means, F-Measure, Sensitivity and Specificity, which are particularly intended for imbalanced classification [19]. The metrics can be described as following:

- **Accuracy:** measures the ratio of correctly known observations of both classes. In fact, it shows the percentage of correct predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

- **Sensitivity:** measures the positives ratio that are correctly recognized (e.g. the percentage of patient STATUE = 1 who are recognized correctly as they have the condition).

$$\text{Sensitivity} = \frac{TP}{FN + TP} \tag{4}$$

- **Specificity:** measures the ratio of negatives that are recognized correctly (e.g. the percentage of patient STATUE = 0 who are recognized correctly as they do not have the condition).

$$\text{Specificity} = \frac{TN}{FP + TN} \tag{5}$$

Where TP is the true positives, FP is the false positives, FN is the false negatives, TN is the true negatives, and $TP + TN + FP + FN = n$ is the overall number of observations.

mmeybodi@aut.ac.ir

- **G-means:** The geometric mean (G-mean) was recommended in [29] as the product of the prediction accuracies for both classes,, i.e. specificity (correctness on the negative samples), and sensitivity (correctness on the positive samples). A poor prediction of the positive class will cause a low G-mean value, even if the negative samples are categorized each model properly [30].

$$\mathrm{G-means} = \sqrt{Specificity * Sensitivity} \tag{6}$$

- **F-Measure:** F-measure is illustrated as the harmonic mean of recall and precision [31]. The value of F-measure is increased considerably in relation to the increased recall and precision. Note that a high value of F-measure represents the better performance of the model on the positive samples.

$$\mathrm{F-Measure} = \frac{2 * Recall * precision}{Recall * precision} \tag{7}$$

4 Results and Discussion

4.1 Role of Attributes and Survival Prediction

The collected dataset from Department of Transplantation and Special Diseases included donor and recipient attributes as well as the pre-transplantation and post-transplantation facts of multi-organs. 578 patients out of 631 patients were alive after liver transplantation without any problem. The post-transplant result of every patient is related to several factors, including the patient's pre-transplantation status, the surgery complication and the graft quality [32]. 631 patients participated in the current study, with a Mean ± standard deviation reception age of 33.308 ± 19.34 years including 393 (62.2%) males and 238 (37.7%) females. Patients have been followed up for a two-year period after the LT to monitor the death possibility, because of liver transplantation complications. In this field, 53 (8.4%) patients died and 578 (91.6%) patients were missed or alive in the follow-up (censored data).

Based on the obtained result, it has been shown that among these 38 factors, 13 factors are the most effective elements regarding to have a better post-transplant outcome by the use of GA, as illustrated in Table 2. In our study, PNF was the important prognosis factor in survival after transplantation; this result is in agreement with most of the previous studies about survival or risk factors of death after transplantation [33]. Probably the direct relation between PNF and complexity of the surgical procedure can increase the risk of death [34]. The results have been shown that all 5 recipients with the graft dysfunction died after LT. Furthermore, the relation between renal failure by increasing the morbidity and mortality rate after heart or liver transplantation in several studies is approved. This is consistent with our findings [35, 36]. Our study includes 39 recipients with renal failure after surgery. In addition, lung complication and acute

mmeybodi@aut.ac.ir

rejection after transplantation in our proposed models are presented as two effective factors on survival which are considered the two common and main causes of death after liver transplantation [37]. The trained model includes the record of 435 recipients with no rejection and 196 patients who have been suffering acute rejection after surgery. Noteworthy, 51 out of 196 died after LT.

It is also found that PTLD is one of the significant parameters for the prediction of survival [38]. The clinical factors such as INR and Total Bilirubin are vital to be considered to predict the result. The obtained INR and Total Bilirubin values in our dataset are 1.98 ± 1.09 and 8.045 ± 10.69, respectively. Prior to surgery, the history of recipient's abdominal surgery is important to assess the overall survival rate after LT. Our model trained 554 recipients with no abdominal surgery, and 77 recipients with previous abdominal surgery.

Additionally, the recipient's age plays an important role for the survival of graft after LT. The age of all recipients was in the range of 33.30 ± 19.34. In our study there were no missing record with a minimum statistic of 2 and maximum statistic of 74 for the recipients. Also, it includes 426 male donors and 205 female donors. All these donors have been used for 393 male and 238 female recipients. Finally, in the current study it has been shown that factors like Cold Ischemia Time and Total Bleeding will affect the survival of patients after the surgery.

4.2 Performance Evaluation

Generally, it is very complicated to define the prediction's accuracy of a patient's survival in a particular medical situation. In this paper, dealing with imbalanced dataset was one of the challenging issues to predict two-year survival. On the other hand, achieving the correct model for evaluation metrics, to some extent, was the main aim of the research. Unbalanced class distribution leads to misreading the common evaluation metrics. Moreover, it will cause a biased classification. As it is mentioned earlier, the best output of the classification models has been obtained by using balancing techniques. Therefore, in our study using a combined measure (G-means, Accuracy and F-Measure) and a graphical performance assessment (Area Under Curve) are proposed for the evaluation of imbalanced data learning. Also in Fig. 2, a comprehensive comparison has been provided of these criteria in examined algorithms.

The performance measures of classification models are evaluated by using ROC curves through WEKA software. During the time of drawing the ROC curves, the Y-axis is the True Positive Rate and X-axis is the False Positive Rate which is illustrated in Fig. 3. The Area under Curve (AUC) of DT and MLP are 0.7530 and 0.7340, respectively. But the AUC of KNN is 0.7183. Researchers proved that the classification models with AUC more than 0.5 can be used for prediction purposes. Thus we can choose DT and MLP for medical prediction purposes.

Furthermore, the obtained results for various measures in classification models are provided in Table 4. Based on the Accuracy criterion as a common parameter to assess the performance of classification models, DT with the accuracy of 80.0% has the highest accuracy among the examined models. Following by that, MLP and KNN are in the next ranks with the accuracy of 77.78% and 70.00%, respectively. Due to the high accuracy, DT is more reliable than others to predict the survival after liver

mmeybodi@aut.ac.ir

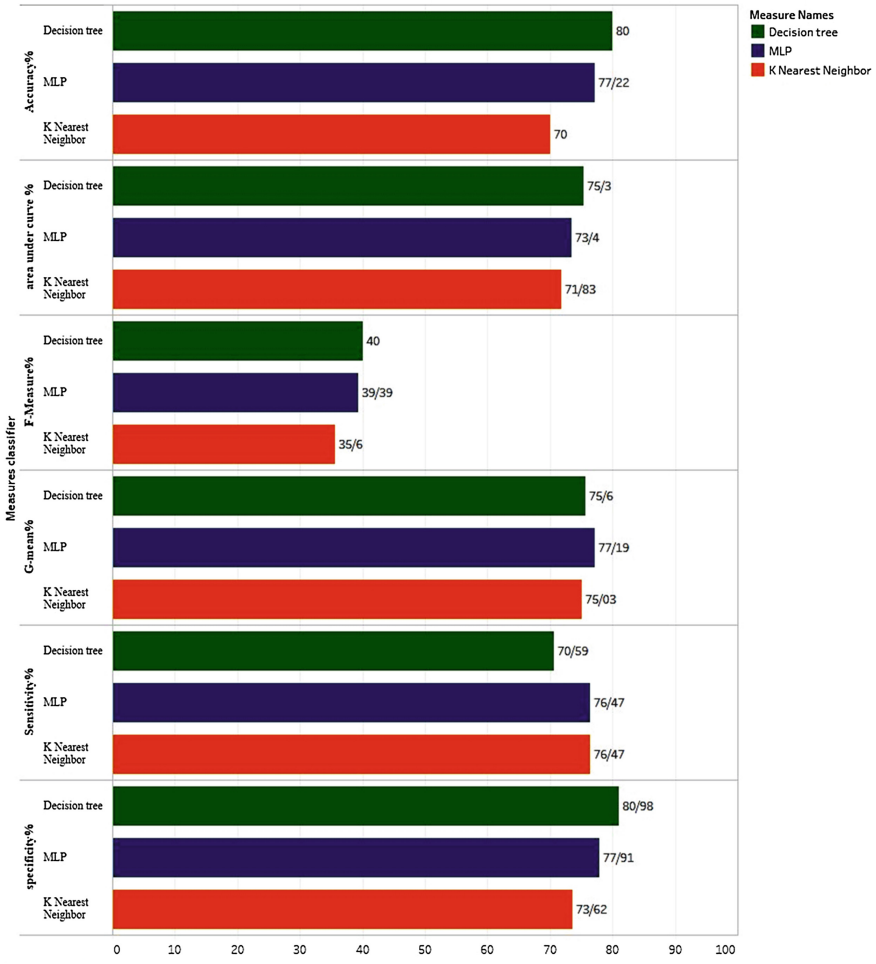


Fig. 2. Comparison of our classifiers based on performance measures.

transplantation as illustrated in Table 4. Three classification models are compared with MELD score. The performance of classification models is good in terms of accuracy, compared with the MELD model. Based on MELD score, the number of survival patients were 151 and 145 patients were dead. Thus the survival rate is 51.01% using MELD score which is lower than that of classification models. But by considering donor recipient and transplantation characteristics, the survival rate of 80.00% is achieved using DT in liver patients after LT.

As stated earlier, as a result of the non-balancing dataset, other measures should be considered to evaluate the proposed algorithms. Correct identification of the patients' percent with liver illness is achieved through MLP and KNN models with 76.47%

mmeybodi@aut.ac.ir

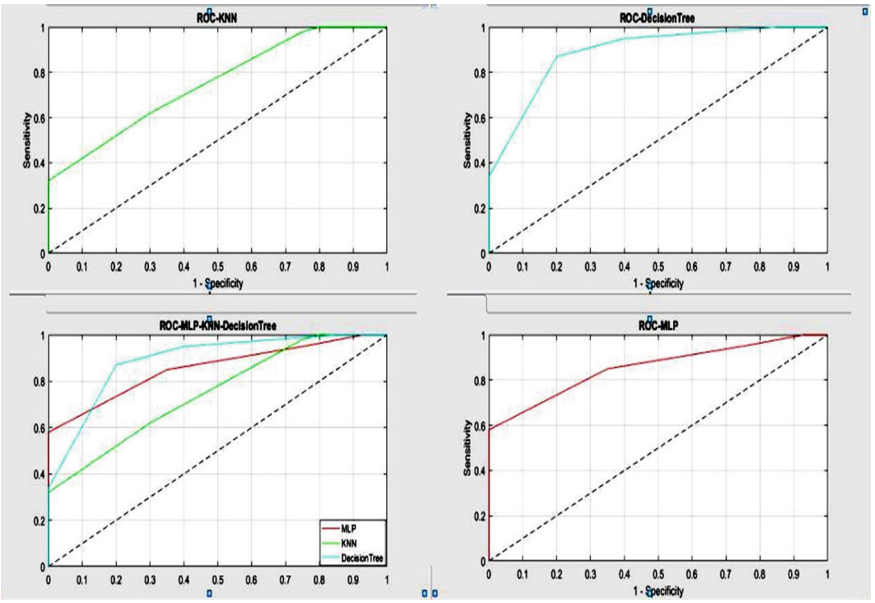


Fig. 3. ROC curves of proposed classification models.

Table 4. A comparison of MLP, Decision tree and KNN in the prediction of liver transplantation patient's survival.

Measures classifier	Accuracy %	Area under curve %	Sensitivity %	Specificity %	F-measure %	G-mean %
MLP	77.78	73.40	77.19	39.39	77.91	76.47
KNN	70.00	71.83	75.03	35.60	73.62	76.47
DT	80.00	75.30	75.60	40.00	80.98	70.59

sensitivity. Moreover, the specificity clearly demonstrated the orientation of different techniques toward the majority class. The whole number of true negatives of the total liver patients shows 80.98% specificity.

4.3 Discussion

Over the past 50 years, LT has been considered the only lifesaving approach for many end stage liver diseases. LT approach was introduced since twenty years ago in Iran, and currently, more than 600 liver transplants are performed per year. The follow up is essential considering the increased number of patients with liver transplant in Iran. Survival prediction is the main factor applied to identify the success of LT surgery. Furthermore, the important attributes influencing the patients' survival after liver transplantation are valuable for pre-operative and post-operative cares. However, a

mmeybodi@aut.ac.ir

small number of studies were conducted on survival in patients with LT so far [39, 40]. As previously mentioned, clinically, patients' survival prediction is based on the MELD score. Using the common statistical techniques is computationally expensive and does not offer reliable results. Data mining techniques, however, provide the flexible and fast solutions with greater and stronger datasets. Moreover, they might be taken into account useful and appropriate tools for the medical prognosis in liver transplantation. Therefore, these models were applied to predict the best liver post-transplantation outcomes of patients by training the given liver dataset, in this study.

The purpose of the present study was to model the survival of patients with LT in an extensive range of age (2 years old and higher) utilizing Artificial Neural Network, Decision tree and K Nearest Neighbor to compare the performance of mentioned models in predicting death caused by the complications of liver transplantation. Based on the obtained results, the accuracy rate of survival prediction was 80% in Decision Tree model (Table 4). The results of our study illustrate that sensitivity was consistent with MLP and KNN models with 76.47%, while specificity was higher in Decision Tree with 80.98%. Furthermore, considering the prominence of the AUC criterion, Decision Tree performs better than other models because of the highest accuracy. Thus, it is clear that Decision Tree performs better to predict the survival of patients after LT.

However, in numerous studies, these techniques were compared for survival analysis in various diseases worldwide [41, 42]. In all these studies, the superiority of data mining techniques mentioned over the conventional statistical techniques in real clinical datasets. A study conducted by Hoot [43] to predict the graft survival rate of liver transplant recipient using ANN. The main limitation of this research was that only fewer attributes have been used and only the accuracy of 67% has been obtained. Brier et al. [44], using ANN and LR by achieving 63% and 64% accuracy respectively, predicted the survival rates. Dorado-Moreno et al. [11] could attain the accuracy of 73% in analyzing survival with imbalance dataset. They utilized ordinary Artificial Neural Network and an ordinary over-sampling technique to alleviate the imbalanced distribution dataset. However, in this research paper, an LT survival prediction deal with the imbalanced nature of the dataset for survival prediction and compared three data mining techniques. Based on the results, the patients' survival of after LT can be predicted with 80% accuracy using Decision Tree model. It is also noteworthy that evaluating the role of numerous different elements in the patients' survival with LT, concurrently and in a real dataset, was another potency of the present study.

5 Conclusions

LT is regarded as an ultimate treatment, particularly for chronic end-stage liver disease. Numerous patients undergo this life-saving treatment because of technological advancements yielding better post-transplantation outcomes, regardless of the expensive nature of the surgery procedure. The medical experts' prediction is on the basis of MELD score, however, the MELD score always will not provide the most accurate outcome. Hence, a number of useful data mining techniques are presented to predict the probability of survival after LT.

mmeybodi@aut.ac.ir

In the current study, the effects of 38 potential attributes on the post-transplantation survival have been considered simultaneously using GA. Moreover, some techniques were applied to tackle imbalanced distributions to avoid errors in prediction models. This approach was taken into account in relation to an under-sampling and over sampling techniques. The best cooperation of these techniques which is being used widely for imbalanced classification is indicated in the study's results. The combination of these two techniques will increase the accuracy of the minority class. To predict post-transplantation survival, MLP, Decision tree and K Nearest Neighbor models are applied. The results of our study represent that the performance of the classification models was substantially improved by balancing the dataset, particularly in the MLP model in the minority class. Through this study, a high accurate survival prediction has been achieved, 80.00%, using Decision Tree model. Thus, our proposed models will become very helpful for the physicians for making precise decisions and predict better post-transplant outcomes. Future research directions can be considered as collecting bigger datasets to make a supranational dataset and to simulate the model in a more controlled situation for analyzing its behavior. In addition, investigating broader data mining techniques or applying the hybrid methods can be aa study, as well.

References

1. Song, A.T.W., Avelino-Silva, V.I., Pecora, R.A.A., Pugliese, V., D'Albuquerque, L.A.C., Abdala, E.: Liver transplantation: fifty years of experience. World J. Gastroenterol. WJG **20** (18), 5363 (2014)
2. Kamath, P.S., Wiesner, R.H., Malinchoc, M., Kremers, W., Therneau, T.M., Kosberg, C.L., D'Amico, G., Dickson, E.R., Kim, W.R.: A model to predict survival in patients with end-stage liver disease. Hepatology **33**(2), 464–470 (2001)
3. Cucchetti, A., Vivarelli, M., Heaton, N.D., Phillips, S., Piscaglia, F., Bolondi, L., La Barba, G., Foxton, M.R., Rela, M., O'Grady, J.: Artificial neural network is superior to MELD in predicting mortality of patients with end-stage liver disease. Gut **56**(2), 253–258 (2007)
4. Su, C.-J., Wu, C.-Y.: JADE implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring. Appl. Soft Comput. **11**(1), 315–325 (2011)
5. Mansingh, G., Osei-Bryson, K.-M., Asnani, M.: Exploring the antecedents of the quality of life of patients with sickle cell disease: using a knowledge discovery and data mining process model-based framework. Heal. Syst. **5**(1), 52–65 (2016)
6. Doyle, H.R., Marino, I.R., Jabbour, N., Zetti, G., McMichael, J., Mitchell, S., Fung, J., Starzl, T.E.: Early death or retransplantation in adults after orthotopic liver transplantation: can outcome be predicted? 1. Transplantation **57**(7), 1028 (1994)
7. Doyle, H.R., Dvorchik, I., Mitchell, S., Marino, I.R., Ebert, F.H., McMichael, J., Fung, J.J.: Predicting outcomes after liver transplantation. A connectionist approach. Ann. Surg. **219**(4), 408 (1994)
8. Marsh, J.W., Dvorchik, I., Subotin, M., Balan, V., Rakela, J., Popechitelev, E.P., Subbotin, V., Casavilla, A., Carr, B.I., Fung, J.J.: The prediction of risk of recurrence and time to recurrence of hepatocellular carcinoma after orthotopic liver transplantation: a pilot study. Hepatology **26**(2), 444–450 (1997)
9. Khosravi, B., Pourahmad, S., Bahreini, A., Nikeghbalian, S., Mehrdad, G.: Five years survival of patients after liver transplantation and its effective factors by neural network and cox proportional hazard regression models. Hepat. Mon. **15**(9), e25164 (2015)

mmeybodi@aut.ac.ir

10. Raji, C.G., Chandra, S.S.V.: Predicting the survival of graft following liver transplantation using a nonlinear model. J. Public Heal. **24**(5), 443–452 (2016)
11. Dorado-Moreno, M., Pérez-Ortiz, M., Gutiérrez, P.A., Ciria, R., Briceño, J., Hervás-Martínez, C.: Dynamically weighted evolutionary ordinal neural network for solving an imbalanced liver transplantation problem. Artif. Intell. Med. **77**, 1–11 (2017)
12. Perez-Ortiz, M., Gutiérrez, P.A., Ayllón-Terán, M.D., Heaton, N., Ciria, R., Briceño, J., Hervás-Martínez, C.: Synthetic semi-supervised learning in imbalanced domains: constructing a model for donor-recipient matching in liver transplantation. Knowl.-Based Syst. **123**, 75–87 (2017)
13. Busuttil, R.W., Tanaka, K.: The utility of marginal donors in liver transplantation. Liver Transplant. **9**(7), 651–663 (2003)
14. Briceno, J., Solorzano, G., Pera, C.: A proposal for scoring marginal liver grafts. Transpl. Int. **13**(1), S249–S252 (2000)
15. Pérez-Ortiz, M., Cruz-Ramírez, M., Ayllón-Terán, M.D., Heaton, N., Ciria, R., Hervás-Martínez, C.: An organ allocation system for liver transplantation based on ordinal regression. Appl. Soft Comput. **14**, 88–98 (2014)
16. Maalouf, M., Siddiqi, M.: Weighted logistic regression for large-scale imbalanced and rare events data. Knowl.-Based Syst. **59**, 142–148 (2014)
17. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **9**, 1263–1284 (2008)
18. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
19. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. Inf. Sci. (Ny) **250**, 113–141 (2013)
20. Yen, S.-J., Lee, Y.-S.: Cluster-based sampling approaches to imbalanced data distributions. In: International Conference on Data Warehousing and Knowledge Discovery, pp. 427–436 (2006)
21. Zhang, Y.-P., Zhang, L.-N., Wang, Y.-C.: Cluster-based majority under-sampling approaches for class imbalance learning. In: 2010 2nd IEEE International Conference on Information and Financial Engineering (ICIFE), pp. 400–404 (2010)
22. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. Evol. Comput. **17**(3), 275–306 (2009)
23. Selvakuberan, K., Indradevi, M., Rajaram, R.: Combined Feature Selection and classification – a novel approach for the categorization of web pages. UK J. Inf. Comput. Sci. **3**(2), 83–89 (2008)
24. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explor. Newslett. **11**(1), 10–18 (2009)
25. Zhang, M., Yin, F., Chen, B., Li, Y.P., Yan, L.N., Wen, T.F., Li, B.: Pretransplant prediction of posttransplant survival for liver recipients with benign end-stage liver diseases: a nonlinear model. PLoS ONE **7**(3), e31256 (2012)
26. Podgorelec, V., Kokol, P., Stiglic, B., Rozman, I.: Decision trees: an overview and their use in medicine. J. Med. Syst. **26**(5), 445–463 (2002)
27. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
28. Santos, M.S., Soares, J.P., Abreu, P.H., Araujo, H., Santos, J.: Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches. IEEE Comput. Intell. Mag. **13**(4), 59–76 (2018)
29. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. ICML **97**, 179–186 (1997)

mmeybodi@aut.ac.ir

30. Hido, S., Kashima, H., Takahashi, Y.: Roughly balanced bagging for imbalanced data. Stat. Anal. Data Min. ASA Data Sci. J. **2**(5–6), 412–426 (2009)
31. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006)
32. Moreno, R., Berenguer, M.: Post-liver transplantation medical complications. Ann. Hepatol. **5**(2), 77–85 (2006)
33. Máthé, Z., Paul, A., Molmenti, E.P., Vernadakis, S., Klein, C.G., Beckebaum, S., Treckmann, J.W., Cicinnati, V.R., Kóbori, L., Sotiropoulos, G.C.: Liver transplantation with donors over the expected lifespan in the model for end-staged liver disease era: is Mother Nature punishing us? Liver Int. **31**(7), 1054–1061 (2011)
34. Sharma, R., Kashyap, R., Jain, A., Safadjou, S., Graham, M., Dwivedi, A.K., Orloff, M.: Surgical complications following liver transplantation in patients with portal vein thrombosis —a single-center perspective. J. Gastrointest. Surg. **14**(3), 520–527 (2010)
35. Tinti, F., Mitterhofer, A.P., Muiesan, P.: Liver transplantation: role of immunosuppression, renal dysfunction and cardiovascular risk factors. Minerva Chir. **67**(1), 1–13 (2012)
36. Santos, C.A.Q., Brennan, D.C., Fraser, V.J., Olsen, M.A.: Incidence, risk factors, and outcomes of delayed-onset cytomegalovirus disease in a large, retrospective cohort of heart transplant recipients. Transpl. Proc. **46**(10), 3585–3592 (2014)
37. Fallon, M.B., Krowka, M.J., Brown, R.S., Trotter, J.F., Zacks, S., Roberts, K.E., Shah, V.H., Kaplowitz, N., Forman, L., Wille, K.: Impact of hepatopulmonary syndrome on quality of life and survival in liver transplant candidates. Gastroenterology **135**(4), 1168–1175 (2008)
38. Dreyzin, A., Lunz, J., Venkat, V., Martin, L., Bond, G.J., Soltys, K.A., Sindhi, R., Mazariegos, G.V.: Long-term outcomes and predictors in pediatric liver retransplantation. Pediatr. Transplant. **19**(8), 866–874 (2015)
39. Cucchetti, A., Piscaglia, F., Grigioni, A.D., Ravaioli, M., Cescon, M., Zanello, M., Grazi, G. L., Golfieri, R., Grigioni, W.F., Pinna, A.D.: Preoperative prediction of hepatocellular carcinoma tumour grade and micro-vascular invasion by means of artificial neural network: a pilot study. J. Hepatol. **52**(6), 880–888 (2010)
40. Ho, W.-H., Lee, K.-T., Chen, H.-Y., Ho, T.-W., Chiu, H.-C.: Disease-free survival after hepatic resection in hepatocellular carcinoma patients: a prediction approach using artificial neural network. PLoS ONE **7**(1), e29179 (2012)
41. Chi, C.-L., Street, W.N., Wolberg, W.H.: Application of artificial neural network-based survival analysis on two breast cancer datasets. In: AMIA Annual Symposium Proceedings, p. 130 (2007)
42. Ansari, D., Nilsson, J., Andersson, R., Regnér, S., Tingstedt, B., Andersson, B.: Artificial neural networks predict survival from pancreatic cancer after radical surgery. Am. J. Surg. **205**(1), 1–7 (2013)
43. Hoot, N.R.: Models to Predict Survival After Liver Transplantation (2005)
44. Brier, M.E., Ray, P.C., Klein, J.B.: Prediction of delayed renal allograft function using an artificial neural network. Nephrol. Dial. Transplant. **18**(12), 2655–2659 (2003)

mmeybodi@aut.ac.ir



Deep Learning Prediction of Heat Propagation on 2-D Domain via Numerical Solution

Behzad Zakeri[1(✉)], Amin Karimi Monsefi[2], and Babak Darafarin[3]

[1] University of Tehran, Tehran, Iran
Behzad.Zakeri@ut.ac.ir
[2] Shahid Beheshti University, Tehran, Iran
A_karimimonsefi@sbu.ac.ir
[3] Amirkabir University of Technology, Tehran, Iran
babak.darafarin@aut.ac.ir

Abstract. Deep learning's role in tackling complicated engineering problems becomes more and more effective by advances in computer science. One of the classical problems in physics is representing the solution of heat propagation in the arbitrary 2-D domain. Study of two-dimensional heat transfer provides a precious bed for related physical issues. In this work, by using finite volume method, we solved the two-dimensional heat equation on the arbitrary domain with specified limitations (considering three heated rectangular obstacles inside the main domain) for 100000 different cases. These cases were divided into big batches in order to reduce the computational cost. The solution for each case was used as sample data to train our deep neural network. After the training process, deep learning results have been compared to results which were produced by the commercial program (ANSYS). After analyzing deep learning efficiency, obviously, our network successfully was able to predict the solution of heat transfer physics with satisfactory precision.

Keywords: Deep neural network · Regularization · Laplace equation · Heat conduction

1 Introduction

Deep learning as a subset of artificial intelligence plays a significant role in various studies, and nowadays, in most of the complicated cases, deep learning is being used to simplify complex computations. Sound recognition [24], pattern recognition [12], and suggestion of relevant topics on the Internet websites [27] are the only small numbers of enormous applications of this powerful tool. The key feature of deep learning is that the layers which are used in learning procedure are not designed by the human, and they perform by using data which is fed to the

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 161–174, 2020.
https://doi.org/10.1007/978-3-030-37309-2_13

mmeybodi@aut.ac.ir

network as an inputs [16]. This feature of deep learning makes it an appropriate choice for representation systematic solution for a wide range of problems with high complexity [2].

On the other hand, Partial Differential Equations as cornerstones of studying dynamical systems, portray a precise image of many natural processes [23]. Although using PDEs to describe natural phenomena is an elegant way of modelling, several factors such as high dimensionality, and complex domains, made it tough to perform analytical solutions in most cases [10]. Representing new methods for solving partial differential equations have always been an interesting subject in computational science [8]. By advances in machine learning and specifically in deep learning the idea of using artificial neural networks for solving differential equations became more favorable [5,33].

The utility of deep learning method compared to other existing numerical methods for solving differential equations has several advantages. Deep learning provides a valuable approach to deal with uncertainties and nonlinearities in differential equations like stochastic PDEs [21]. Furthermore, it is noticeable that deep learning method is a perfect tool to prevent instability in solving procedure. Instability in other numerical methods is a big concern and in some cases caused divergence in solving procedure [1]. In deep learning algorithms, the stability of the solution highly dependent on weights W which are chosen by solver itself [25]. The heat equation is one of the most important equations in mathematical physics and engineering [22]. Heat conduction has been modelled by Joseph Fourier using the second-order partial differential equation [9]. Two-dimensional heat conduction (known as ***Laplace*** equation) in the rectangular domain is one of the famous classical problems in engineering mathematics and heat transfer, and there are extended numerical and analytical solutions for it [3,14]. However, if the domain of the solving equation changes from rectangular to an arbitrary domain, it will be really tough to represent the analytical solution for that case, and the only way would be using numerical methods [19].

We are interested in establishing a new method for studying natural phenomena which are modelled by PDEs, specifically problems related to fluid dynamics and heat transfer. In this paper, we focus on the case of 2-dimensional steady state heat propagation inside the rectangular domain considering three smaller rectangular obstacles with different temperatures inside it. We solved the governing equation for a large number of cases with different positions, sizes and temperatures in obstacles by taking advantage of the finite difference method. By changing the conditions of the problem and providing various situations, we provide sufficient labelled inputs for our learning algorithm.

The fully connected deep neural network has been designed with several hidden layers for this case. Since the number of input data was extremely large, over-fitting methods have used to prevent this phenomenon in the learning

mmeybodi@aut.ac.ir

procedure. After the learning process, the outputs of the network compared to results for the same conditions which were produced by commercial software for solving heat transfer and fluid flow problems, ANSYS FLUENT 19.0. Moreover, in the specified case with no heated obstacle, the deep learning result has been compared to the analytical solution extracted by orthogonal functions using Fourier series.

2 Related Work

This work is a combination of two separate fields of study pursued by a wide range of research communities. In this perspective, it is tried to find the best and optimum deep learning algorithm to find the solution of the specified heat conduction problem by taking advantage of the finite volume method in order to generate learning data.

Laplace equation plays an important role in many scientific fields, such as complex analysis [29], electromagnetic fields [34], fluid flow and heat transfer. One of the first successful tries for solving this equation on the arbitrary domain using numerical methods has been performed by Bruch and Zyvoloski for heat conduction purposes in 1974 [4]. Although mesh based numerical methods are strong tools in engineering problems, stability, the dependency of the solution to the mesh and Necessity of resolving the problem by changing the conditions of the problem are disadvantages of these methods, and motivate scientists to search for analytical solutions or at least mesh-less methods [17]. Many efforts went on finding an analytical solution for the Laplace equation on the arbitrary domain. Crowdy represent an analytical solution for potential flow (Laplace equation) past through obstacles on the infinite domain [6]. However, his attempts cannot solve the same problem for heat propagation in a finite domain, because of the difference in the boundary conditions.

Deep learning as an intelligence tool for prediction of the behaviour of the dynamical systems widely has been used in thermal-fluid sciences. Miyanawala and Jaiman has conducted an efficient deep learning technique for the model reduction of the unsteady Navier-stocks equation flow problems [20]. Several other types of research have been done related to the simulation and prediction of the fluid flow dynamics [11,13,18,31].

Since predicting the solution of differential equations using deep neural networks requires having a large number of labelled correct input data, weakly supervised learning algorithm using appropriate chosen convolutional kernel would be could be a good choice for simple cases. This method can learn directly from the initial condition [26]. Although this technique (which known as the physical informed network) predict the solution with good accuracy for simple physics, we focus on conventional learning methods to study the accuracy of these methods in dealing with such problems.

mmeybodi@aut.ac.ir

3 Methodology

This section is divided into two main parts. Firstly, the physics of heat propagation in a two-dimensional domain (Ω) explain briefly, and also it is described how input data for the learning procedure have been generated. In the second part, the deep learning approach and our algorithms are discussed.

3.1 Heat-Transport

Looking at details of two-dimensional heat propagation mathematics, the first step in this part is defining a proper space for solving the governing equation. We have decided to choose the square domain for solving the Laplace equation. Each side of the domain has its own boundary condition. Inside the main square, three heated rectangular obstacles with arbitrary sizes and positions have been considered Fig. 1. The aim of this definition is training our deep neural network to learn the pattern of heat propagation, and make it able to predict the correct temperature contours when the user gives the boundary conditions, positions and sizes of the obstacles.

It is important to note that because of the high computational cost, we consider the same temperature for all three obstacles, although this assumption can be changed easily.

The general heat conduction formation known as 2-D transient heat conduction can be written in the form of Eq. 1 :

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \tag{1}$$

where $\alpha > 0$ is a coefficient of the thermal diffusivity of the plate, and $T = T(x, y, t)$ demonstrates the temperature value in the given position and time.

However, in this study, we are interested in studying the steady state of heat transfer. So, the Eq. 1 would reform to Eq. 2 :

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \tag{2}$$

To solve Eq. 2 we need to specify the boundary conditions of the problem.

To solve the equation. in the simple rectangular domain with simplified boundary conditions, there are several analytical methods, such as separation of variables and using the error function. However, these methods in dealing with more complicated B.C or domains become useless, and it is necessary to use numerical methods.

mmeybodi@aut.ac.ir

In this work, we considered constant temperature on our boundaries which is known as the Dirichlet boundary condition [7].

Equations 3 and 4 depict the boundary conditions definition as follows :

$$\partial\Omega = \sum_{i=1}^{4} \partial T_i + \sum_{s=1}^{3} \partial A_s \tag{3}$$

$$T|_{\partial\Omega} = cte \tag{4}$$

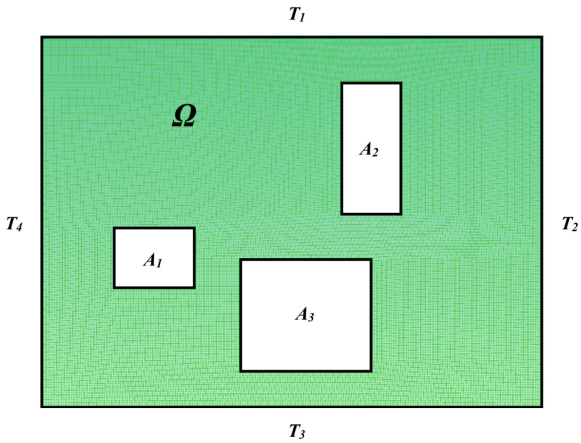


Fig. 1. Sample domain with obstacles

3.2 Data Generation

In order to provide proper input data as nourishment of the deep learning algorithm, the Laplace equation has been solved numerically for various conditions. These data after some treatment have been used in the input layer to train the network correctly.

Finite Volume Method. There are several numerical methods which iteratively solve equations which are not possibly solved by analytical methods. Finite Volume is one of the comprehensive methods that can deal with complex problems in solving differential equations. Although the concept of finite volume is based on 3-D problems, it can easily be extended to less topological dimensions [32].

To solve the Laplace equation using FVM, we need to discretize $\nabla^2 T = 0$. The temperature of the node (i, j) Fig. 2 calculates as follows :

$$\int_{\Delta V} \frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right) dx.dy + \int_{\Delta V} \frac{\partial}{\partial y}\left(\frac{\partial T}{\partial y}\right) dx.dy = 0 \tag{5}$$

mmeybodi@aut.ac.ir

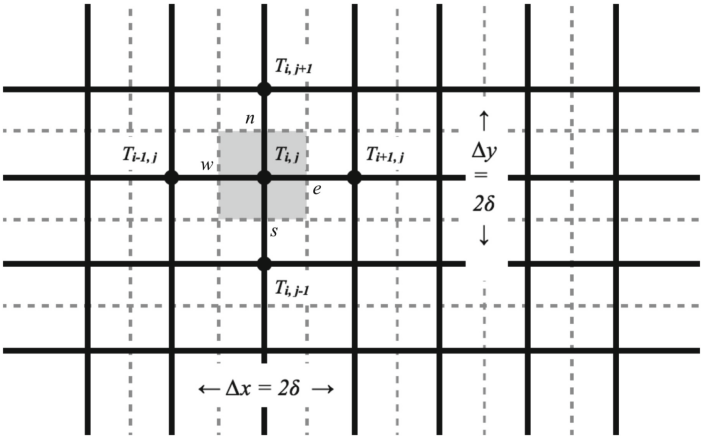


Fig. 2. Discretization of the domain

With assuming uniform square mesh and also considering linear temperature flux change along directions calculation continues as follows :

$$\Delta y = \Delta x \rightarrow A_e = A_w = A_n = A_s \tag{6}$$

$$\Gamma = \frac{A}{\delta} \tag{7}$$

$$4\Gamma T_p = \Gamma(T_w + T_s + T_e + T_n) \tag{8}$$

Based on Eq. 8, temperature of the node (i, j) can be calculated by Eq. 9 :

$$T_{i,j} = \frac{T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{4} \tag{9}$$

Equation 9 was solved iteratively with Dirichlet boundary condition until convergence.

Input Data Preparation. For easier analysis of produced data, we divide the solution of the Eq. 9 into 40 big batches. Each batch contains input and output files. The input file has been performed by 2500 combination of 19 separate elements, such as the width and height of the main domain, size and position of each rectangular obstacle, and also temperatures of each side of the domain. For each set of input elements, a specified solution has been assigned using Eq. 9.

mmeybodi@aut.ac.ir

Algorithm 1 demonstrates the procedure of solving the Eq. 9 for each input matrix by assuming discussed conditions.

Input:
width, height, top_temperature, right_temperature, left_temperature
bottom_temperature
first_rectangle, second_rectangle, third_rectangle, fixed_temperature
Result:
Temperatire Distribution
Initialization:
$\{T_{i,j}\}_{i=1,j=1}^{width,height} \leftarrow 0$
$\{T_{1,j}\}_{j=1}^{height} \leftarrow top_temprature$
$\{T_{i,height}\}_{i=1}^{width} \leftarrow right_temprature$
$\{T_{i,1}\}_{i=1}^{width} \leftarrow left_temprature$
$\{T_{width,j}\}_{j=1}^{height} \leftarrow bottom_temprature$
$SetFixedTempratureInRectangle(T, first_rectangle, fixed_temprature)$
$SetFixedTempratureInRectangle(T, second_rectangle, fixed_temprature)$
$SetFixedTempratureInRectangle(T, third_rectangle, fixed_temprature)$
$dt \leftarrow 0.25$
$TOL \leftarrow 1e-6$
while $error > TOL$ **do**
  tmp ← T
  for $i \leftarrow 1$ *to width* **do**
    for $j \leftarrow 1$ *to height* **do**
      if $\neg PointIsInRectangles(T_{i,j})$ **then**
        $tmp_x \leftarrow tmp_{i+1,j} - 2 * tmp_{i,j} + tmp_{i-1,j}$
        $tmp_y \leftarrow tmp_{i,j+1} - 2 * tmp_{i,j} + tmp_{i,j-1}$
        $T_{i,j} \leftarrow dt * (tmp_x + tmp_y) + tmp_{i,j}$
      else
        continue
      end
    end
  end
  $error \leftarrow Max(Abstract(Subtract(tmp, T)))$
end

Algorithm 1. Numerical data generation algorithm

Data Treatment. To prepare inputs of our deep neural network, firstly, by taking advantage of average and variance, our data has been normalized. Then, each element of the input matrices which indicates 19 initial data and address of that element (i and j) will be considered as input features for the neural network.

mmeybodi@aut.ac.ir

The output of the deep learning network will be compared to the solution for the corresponding element which is extracted from the output file.

In order to ensure that our network will not be biased by a small proportion of matrices, we have considered an ***acceptance rate*** to guarantee that no more than a specified percentage of elements will be picked from a certain matrix.

3.3 Deep Learning

Deep learning is formed by three main part which are the input layer, hidden layer and output layer. The input layer is a port for importing data into the network. These data have been sent to the network in matrix form. In this study, by using 21 neurons data was transferred from the input layer to the hidden layer. Hidden Layer contains several sublayers, and each of them is made by the specified number of neurons. This stage as the main part of the learning procedure should learn the way that our certain physics work and predict the correct temperature distribution. Finally, the output layer reports the results to the user.

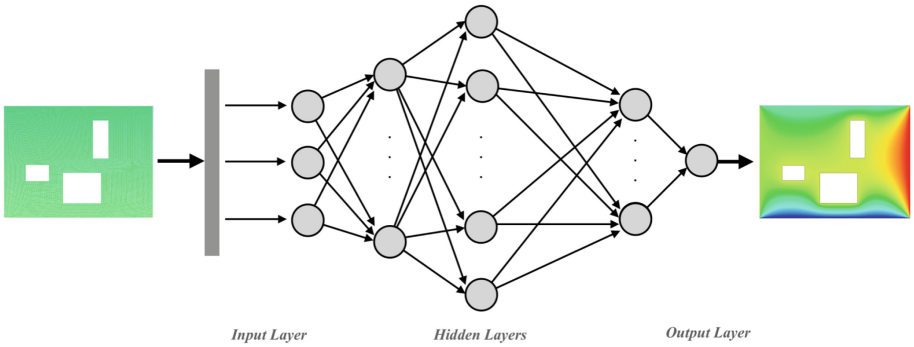


Fig. 3. Deep neural network diagram

Figure 3 illustrates the architecture of the deep learning process. In this architecture, the hidden layer consists of L layers. The schematic function of each neuron in the hidden layer can be shown as Fig. 4. Input data for each neuron receives from all neurons in the previous layer. These inputs by using vector weight (W) and the bias value (B), linearly combined ($\overrightarrow{W}X + B$) and the output result for the neuron is calculated. The process at each neuron will get finished by implementing the activation function. In this stage we used the ***Leaky Relu*** activation function as shown in Eq. 10 :

$$Leaky\text{Re}lu(x) = \begin{cases} x, & x > 0 \\ x * 0.01, & x \leq 0 \end{cases} \tag{10}$$

mmeybodi@aut.ac.ir

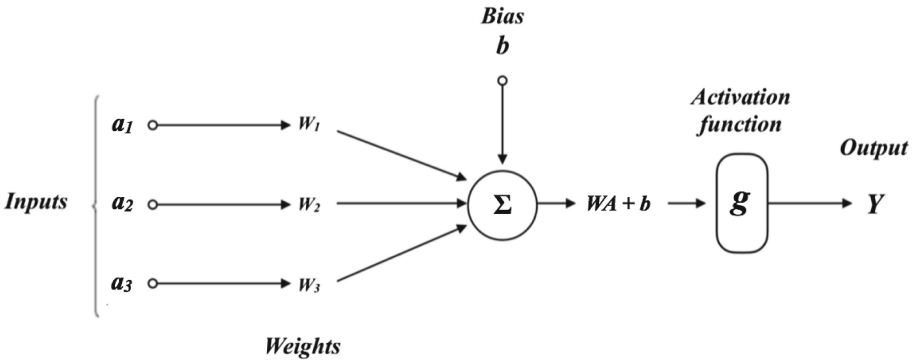


Fig. 4. Single neuron diagram

In general for layer l, according to the Fig. 4 output of the layer l is equal to $Z^{[l]}$ which is shown in Eq. 11 :

$$Z^{[l]} = W^{[l]} * A^{[l-1]} + B^{[l]} \tag{11}$$

Where $w^{[l]}$ is the weight matrix of input for layer l, $A^{[l-1]}$ is input of the layer, and $B^{[l]}$ is a vector of bias values of this layer.

Also, the input of the layer $A^{[l]}$ is defined as follows :

$$A^{[l]} = g^{[l]}(Z^{[l]}) \tag{12}$$

The function $g^{[l]}$ in Eq. 12 represent the activation function in layer l.

Before starting the learning procedure, the values of the $B^{[l]}$ are 0, and the elements of the matrix $W^{[l]}$ are initialized randomly between 0 and 1.

The purpose of the learning network is finding proper w and B for each layer which minimizes the error function.

$$\min_{W,B} J(W, B) \tag{13}$$

Where J is error function which is defined as follows :

$$J(W, B) = \frac{1}{m} \|Y' - Y\|_2^2 \tag{14}$$

In Eq. 14, Y' demonstrates amount of data which is generated by deep learning, also Y and m are real data and numbers of input data respectively.

To prevent over fitting three regularization techniques which are ***Dropout*** [28], ***Momentum*** [30] and ***Weight decay*** [15] have been utilized simultaneously.

mmeybodi@aut.ac.ir

After implementation of these three methods Eq. 14 reformed to Eq. 15 as follows :

$$J(W,B) = \frac{1}{m} \left\| Y' - Y \right\|_2^2 + \frac{\lambda}{2m} \left\| W \right\|_2^2 \tag{15}$$

Where λ is a coefficient which should be set in a way that minimizes the error function.

There are several optimization methods to minimize error function 15. In this work, we checked different optimizers to get the best accuracy, and finally, we chose ***SGD*** (Stochastic Gradient Descent) as our optimizer function. This algorithm, by updating the parameters θ_n of the object $J(\theta_n)$ (as shown in Eq. 16) tries to find the best parameters for minimizing the error function.

$$\theta_{n+1} = \theta_n - \alpha \frac{\partial}{\partial \theta_n} J(\theta_n) \tag{16}$$

In Eq. 16 θ is a vector parameter, also J and α are cost function and slope parameter respectively.

The *SGD* algorithm can estimate the gradient of the parameters only by using a limited number of training examples.

Finally, to find out the learning parameters we categorize the generated data into 3 main categories. From all generated data, 98% has been allocated for training, and the percentage of validating and testing was 1% for each. Also, for more precision and less run time, training data were divided into 1000 mini-batches.

4 Results

In this section, results which have been generated by deep learning is compared to true data by taking advantage of different experiments. In the first stage, deep learning's was analyzed based on the error rate in training and test time. The next step was comparing deep learning results by ANSYS answers. And finally, the accuracy of our network was analyzed by the utility of the analytical solution for the simplified case.

4.1 Analyzing Deep Learning Results

In this section, deep learning precision was analyzed by changes the number of epochs and varying threshold coefficient. For this purpose, we used a different number of epochs (from 100 to 2000) in the input layer. Also by changing the threshold coefficient, it is possible to monitor the effect of epochs in the final results. In this experiment, 98% of true data was considered for training the network, and 2% for validation and test.

mmeybodi@aut.ac.ir

The *Mean Square Error* index has been used to calculate the training and test error. This index is defined as follows 17 :

$$MSE = \frac{\sum_{i=0}^{n} \left(y^i - y'^i\right)^2}{n} \tag{17}$$

The *Threshold* concept has utilized in order to compare the true data with the results from the deep learning method. Whereas y' is the deep learning calculated quantity and y represents the amount of numerical solution generated by FVM.

If the threshold quantity was more than left-hand side of Eq. 18, then both values will be assumed as equal.

$$|y - y'| < \theta \tag{18}$$

Table 1. Epochs' number effect on deep learning results

Epoch number	Training error	Test error	*Th(1)*	*Th(0.1)*	*Th(0.01)*
100	0.984	4.125	75.12%	71.78%	67.47%
200	0.876	3.745	77.83%	74.34%	69.87%
300	0.821	3.424	79.47%	77.54%	72.39%
500	0.700	2.145	87.08%	83.75%	80.19%
1000	0.576	1.406	94.19%	91.07%	89.49%
2000	0.319	0.958	97.19%	93.67%	91.87%

Looking at the Table 1 in more detail, clearly, by increasing the epoch number, precision of the final results was increased for all thresholds. Also, by considering an epoch number, the precision decrease in smaller thresholds.

4.2 Comparison with ANSYS

For engineering purposes, we need to visualize the results of computations to make it easier for engineers to have better judgment about them.

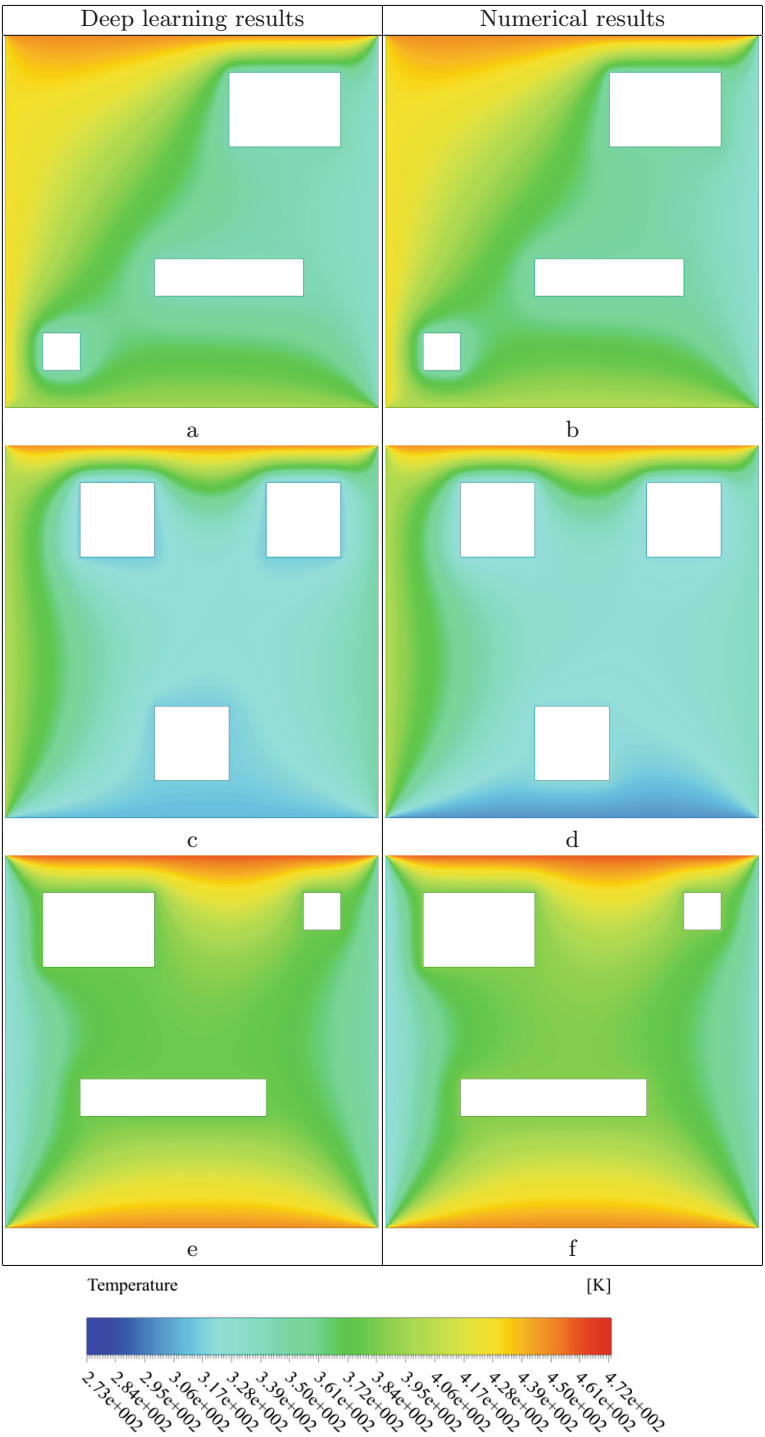
In this part, we compare the results which are extracted by deep learning algorithm with the output of the commercial program (ANSYS Fluent 19.0).

Same geometries with high-quality mesh have been generated and imported to the Fluent solver. All the computations have conducted by the second-order scheme, and the calculations have proceeded until the full convergence.

In Table 2 three sample cases of deep learning and numerical results are compared. Although ANSYS results were quite similar to deep learning output, in regions that thermal gradient was more than other areas, deep learning could not perfectly estimate the temperature distribution.

mmeybodi@aut.ac.ir

Table 2. Comparing deep learning with ANSYS results



mmeybodi@aut.ac.ir

5 Conclusion

We have shown that deep learning successfully can learn the physics of heat transfer in two-dimensional space. We found that there are various factors which directly influence the quality of the deep learning prediction, such as optimizer method, activation function and momentum variable. It is found that the stochastic gradient descent obviously has better performance in comparison to other optimizers. Our deep learning results sufficiently were similar to ANSYS results considering the number of data which were utilized for training the network. Overall, deep learning as a strong tool can provide an amazing method for representing the numerical solution for different kinds of PDEs.

References

1. Ascher, U.M.: Numerical Methods for Evolutionary Differential Equations. vol. 5. Siam (2008)
2. Baker, B., Gupta, O., Naik, N., Raskar, R.: Designing neural network architectures using reinforcement learning. arXiv preprint arXiv:1611.02167 (2016)
3. Bergman, T.L., Incropera, F.P., Lavine, A.S., Dewitt, D.P.: Introduction to Heat Transfer. Wiley (2011)
4. Bruch Jr., J.C., Zyvoloski, G.: Transient two-dimensional heat conduction problems solved by the finite element method. Int. J. Numer. Methods Eng. **8**(3), 481–494 (1974)
5. Chakraverty, S., Mall, S.: Artificial Neural Networks for Engineers and Scientists: Solving Ordinary Differential Equations. CRC Press (2017)
6. Crowdy, D.G.: Analytical solutions for uniform potential flow past multiple cylinders. Eur. J. Mech. B/Fluids **25**(4), 459–470 (2006)
7. Dirichlet, P.G.L.: Über einen neuen Ausdruck zur Bestimmung der Dichtigkeit einer unendlich dünnen Kugelschale, wenn der Werth des Potentials derselben in jedem Punkte ihrer Oberfläche gegeben ist. Dümmler in Komm (1852)
8. Fan, E.: Extended tanh-function method and its applications to nonlinear equations. Phys. Lett. A **277**(4–5), 212–218 (2000)
9. Grattan-Guinness, I., Fourier, J.B.J., et al.: Joseph Fourier, 1768-1830; a survey of his life and work, based on a critical edition of his monograph on the propagation of heat, presented to the Institut de France in 1807. MIT Press (1972)
10. Han, J., Jentzen, A., Weinan, E.: Solving high-dimensional partial differential equations using deep learning. Proc. Nat. Acad. Sci. **115**(34), 8505–8510 (2018)
11. Jeong, S., Solenthaler, B., Pollefeys, M., Gross, M., et al.: Data-driven fluid simulations using regression forests. ACM Trans. Graph. (TOG) **34**(6), 199 (2015)
12. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
13. Kim, B., Azevedo, V.C., Thuerey, N., Kim, T., Gross, M., Solenthaler, B.: Deep fluids: a generative network for parameterized fluid simulations. arXiv preprint arXiv:1806.02071 (2018)
14. Kreyszig, E.: Advanced Engineering Mathematics. Wiley (2010)
15. Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. In: Advances in Neural Information Processing Systems, pp. 950–957 (1992)

mmeybodi@aut.ac.ir

16. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
17. Li, H., Mulay, S.S.: Meshless Methods and Their Numerical Properties. CRC Press (2013)
18. Ling, J., Kurzawski, A., Templeton, J.: Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. J. Fluid Mech. **807**, 155–166 (2016)
19. Minkowycz, W.: Advances in Numerical Heat Transfer. vol. 1. CRC Press (1996)
20. Miyanawala, T.P., Jaiman, R.K.: An efficient deep learning technique for the Navier-Stokes equations: application to unsteady wake flow dynamics. arXiv preprint arXiv:1710.09099 (2017)
21. Nabian, M.A., Meidani, H.: A deep neural network surrogate for high-dimensional random partial differential equations. arXiv preprint arXiv:1806.02957 (2018)
22. Narasimhan, T.: Fourier's heat conduction equation: history, influence, and connections. Rev. Geophys. **37**(1), 151–172 (1999)
23. Robinson, J.C.: Infinite-Dimensional Dynamical Systems: An Introduction to Dissipative Parabolic PDEs and the Theory of Global Attractors. vol. 28. Cambridge University Press (2001)
24. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
25. Ruthotto, L., Haber, E.: Deep neural networks motivated by partial differential equations. arXiv preprint arXiv:1804.04272 (2018)
26. Sharma, R., Farimani, A.B., Gomes, J., Eastman, P., Pande, V.: Weakly-supervised deep learning of heat transport via physics informed loss. arXiv preprint arXiv:1807.11374 (2018)
27. Singhal, A., Sinha, P., Pant, R.: Use of deep learning in modern recommendation system: a summary of recent works. arXiv preprint arXiv:1712.07525 (2017)
28. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
29. Stewart, I., Tall, D.: Complex Analysis. Cambridge University Press (2018)
30. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: International Conference on Machine Learning, pp. 1139–1147 (2013)
31. Tompson, J., Schlachter, K., Sprechmann, P., Perlin, K.: Accelerating eulerian fluid simulation with convolutional networks. arXiv preprint arXiv:1607.03597 (2016)
32. Versteeg, H.K., Malalasekera, W.: An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Pearson Education (2007)
33. Yadav, N., Yadav, A., Kumar, M.: An Introduction to Neural Network Methods for Differential Equations. Springer (2015)
34. Zhang, K., Li, D., Chang, K., Zhang, K., Li, D.: Electromagnetic Theory for Microwaves and Optoelectronics. Springer (1998)

mmeybodi@aut.ac.ir

Cluster Based User Identification and Authentication for the Internet of Things Platform

Rafflesia Khan(✉) and Md.Rafiqul Islam

Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh
rafflesiakhan.nw@gmail.com, dmri1978@gmail.com

Abstract. Data security is very important in Internet of Things (IoT) based system. One of the main issues of security is proper and secure identification and authentication of users in an IoT environment. In this paper, we propose a cluster-based identification and authentication model for IoT platform. The contribution of this research work includes a dynamically configurable system framework that is capable of ensuring identification and authentication to every connected device in IoT platform regardless of their type, location and different parameters. The proposed mechanism ensures a central defense for every IoT service deployed locally or in the cloud by identifying and authenticating smart objects using cluster based identification and authentication process. This cluster based process makes our proposed system a more robust and scalable system architecture that supports both limited-resource and ensemble devices. Eventually, it ensures a continuous secure communication among all identified and authorized cluster members and also continuously prohibits all unauthorized members from causing any interruption. Finally, we have presented a comparative analysis of the performance and effectiveness of the proposed system which reflects different significant capabilities of the work.

Keywords: Internet of Things (IoT) · Security · Threat · Identification · Authentication · Dynamic configuration · Access permission

1 Introduction

With the development of IoT, a huge number of physical devices are interrelated using different networking protocols which enable these IoT-devices or IoT-agents to share resources over the network and also to exchange data, resources and control instructions among them. The history of IoT research, proposed by Ashton [1], dates back to the 1999s. And over the last decade, the research interest around this concept has experienced exponential growth among both research-communities and industries. In recent time, any physical object can be

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 175–187, 2020.
https://doi.org/10.1007/978-3-030-37309-2_14

mmeybodi@aut.ac.ir

transformed into an IoT device/agent if it can be connected to the internet and controlled that way, so there are already more connected devices than people in the world. According to analysts, this will likely reach 20.4 billion by 2020 [2]. This impressively big number of IoT technologies, implemented within our territory, are continuously sharing data as well as information and are encroaching on every aspect of our lives, including our homes, offices, cars and even our bodies. Figure 1 shows a simple scenario of how IoT is surrounding a single human's everyday life.

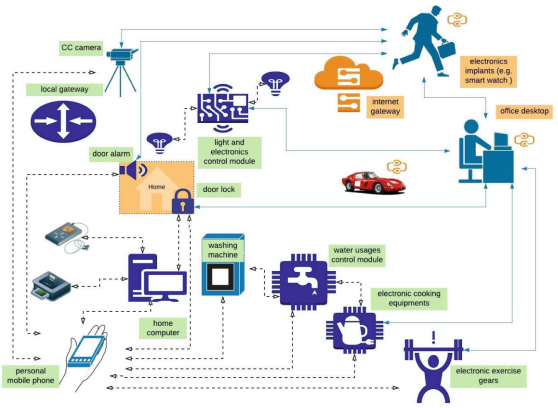


Fig. 1. A scenario of IoT involvement in human life.

Since the IoT devices directly or indirectly have a great impact on the lives of its users, we must give higher priority in order to ensure the security of every device as well as it's user. And there must be some proper well-defined security infrastructure with new technology strategies and protocols that can limit the possible threats related to security challenges. IoT security challenges include different aspects like identification, authentication, privacy, trustworthiness, scalability, availability, confidentiality and integrity. To design a system that can combine all these together is quite difficult and considerably less efficient up to now. So in this paper, we are considering identification and authentication (I&A) as our main concern. Now-a-days IoT based systems like smart city, education, billing system, transportation and governance etc are very complex ones and they use many sensitive data. Security of these sensitive data is very important issue for a complex IoT based system. Because of possibility of the presence of malicious users, security of the data can not be maintained if the users are not properly identified and authenticated. Considering the importance of these kinds of IoT security, recently user identification and authentication for IoT is receiving a lot of attention within the information-security engineers and research communities.

Within an IoT paradigm, devices of different construction, application and characteristics remain interconnected and share confidential information. So,

mmeybodi@aut.ac.ir

identifying (e.g. checking whether a user is valid or not) and authenticating (e.g. checking the identity claim presented by the user) each and every connected device accurately is a major prerequisite. Within an IoT connection, identification of every device as well as user helps every user to identify secure devices and at the same time prohibit insecure devices to establish a connection. On the other hand, authentication can prevent unauthorized users from gaining access to resources and at the same time help legitimate users to access resources in an authorized manner. So, a mutual identification and authentication is highly important and needed because every user within IoT connection needs to be sure of the legitimacies of all the entities involved. Considering this significant importance many recent researchers are working for establishing a mutual and continuous identified and authenticated secure channel between every entity in IoT. Among these existing works some provide I&A service for only local users or devices [3] but for enjoying the benefit of IoT communication we need to identify as well as authenticate each and every local and global device as well. Work like [4], is a very interesting approach but at the same time is computationally expensive as it provides authentication and privacy using IPsec and TLS. Analyzing some survey reports like [5,6] we have identified some still existing issues, challenges, and directions for ensuring a better as well as efficient identification and authentication mechanism. Among them, one of the challenges states that IoT is comprised of a huge number of diverse objects and so to design efficient mechanisms for identifying and authenticating each device and the associated objects used for those devices is very complex and difficult. Also, different objects have different kinds of associated data which are heterogeneous and has no common structure. Because of these reasons, a unified identification and authentication service cannot often be helpful. Therefore, a dynamically configurable service for identification and authentication is needed that can be configured by different kind of objects as well as different data types and can be used comprehensively.

Considering all the above mentioned challenges, in this paper, we are proposing a dynamically configurable cluster based architecture for ensuring user's I&A for IoT environment, which we have referred as I&AIoT. This system can be configurable by every kind of IoT connected devices. Also, we have designed this system as a cloud based service therefore, it would not use resources from local device and so limited-resource or small-scale devices would no longer be a threat for high performance identification procedure. The main contributions of our proposed work are as follows:

- First, to ensure one of the main security issues of IoT, (e.g. identification and authentication) we have presented a cluster based architecture for our proposed IoT service.
- Second, we have designed our system as a cloud based service so that this single identification and authentication service can authenticate both secure and insecure subject and object to ensure security over global IoT paradigm.

mmeybodi@aut.ac.ir

– Third, we have provided a description of the working principle of our model and explained how this system actually assures identification and authentication for IoT in different cases.
– Fourth, we have evaluated the performance of our model in an efficient way that distinctly shows how our model can be a better contribution to IoT security.

2 Related Works

Considering the immense importance, the works on identification and authentication schemes for IoT have been growing rapidly, aiming to address the emerging I&A issues and challenges surrounding IoT applications. In this section, we provide an overview of how researchers have been addressing I&A threats regarding different aspects of IoT.

A recent model based on SDN presents an identification and authentication scheme for heterogeneous IoT networks [7]. This model is based on virtual IPv6 addresses which authenticates devices and gateways, also here different technology-specific identities from different silos are translated by the central SDN into a shared identity. Shivraj et al. proposed an efficient and secure One Time Password technique that is developed with Elliptic Curves Cryptography [8] where Key Distribution Center does not store private and public keys of devices, it only stores their IDs. Dynamic authentication protocol is another interesting project where the time generated by every device is hashed first and then used for identification of the associated device [9]. Sungchul et al. developed an authentication technique [10] that uses the URIs as unique IDs for generating the keys using ECC on an ID-based authentication (IBA) scheme in the context of RESTful web services. Another authentication scheme, useful for limited-capability having things for IoT is proposed in [11] which is based on the association of things with a registration authority. A number of existing models like [12] think that mutual authentication using RFID tag is the most common and easy way to secure IoT devices from encroachment and ensure better data integrity and confidentiality. But this kind of schemes mostly have limited computation and storage capabilities. In [13], a yoking-proof-based authentication protocol (YPAP) has been proposed for cloud-assisted wearable devices. Here yoking-proofs are established for the cloud server to perform simultaneous verification and to realize mutual authentication between a smartphone and two wearable devices lightweight cryptographic operators and a physically unclonable function are jointly applied. But IoT is not limited within some wearable devices.

Analyzing some of states of the arts we have come up with a conclusion that there still exists the necessity of a single, simple and efficient IoT identification and authentication ensuring service, that can serve every kind of data and data containing devices within minimum cost of power and memory space. So we need an efficient identification and authentication (I&A) ensuring model that can be easily configurable and usable by any kind of device and also it should be a cloud

mmeybodi@aut.ac.ir

based service so that it can provide service both locally and globally. Also, by being a cloud based service, it will remain lightweight as well as scalable and appropriate for many resource-limited and small-scale IoT devices.

3 Cluster Based Authentication

For our proposed model, we use the cluster-based authentication process. The system creates some clusters for secure communication. Each cluster is made with the subjects/users of similar functionalities or user habits or related application(s) they are accessing. A node in a cluster is called a cluster member. There is a special member called cluster-master or master which works as a master of all the corresponding members of that class. A member of the cluster is denoted as m_{ij} which represents that m is the ith member of the jth cluster of the system. The master is denoted as M_j that represents the cluster-master of the jth cluster. Since there will be several clusters in a system of the IoT environment, we are using such representations. If there are maximum x members in each cluster and y clusters in the system, $1 \leq i \leq x$ and $1 \leq j \leq y$. A cluster with it's master and members is depicted in Fig. 2.

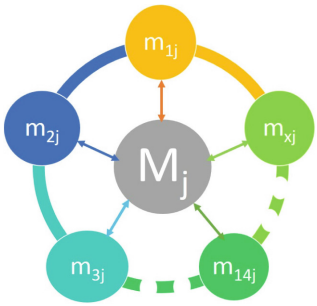


Fig. 2. A cluster j with it's members.

In this model, each cluster member stores its ID and password (pwd), which will be used for generating its private key. We assume a cloud based IoT environment where the system generates a private key for each cluster member and stores it in a private key table with respect to the cluster member's number or id. The private key is generated by a cryptographic hash function such as SHA 256. Next, for each member there is an authentication key, which is generated by the system as follows:

$$K_{auth,ij} = H(ID_{ij}||K_{pr,ij}||CS_j) \tag{1}$$

where i and j represent the member number and cluster number respectively. $K_{pr,ij}$ is the private key for *ith* member in *jth* cluster, CS is the cluster secret

mmeybodi@aut.ac.ir

which is stored in the master of the respective cluster. Each member will be authenticated by this authentication key. If a member (m_{1j}) originates a message and wants to send it to another individual member (m_{2j}), both of the members need to be authenticated. Before sending the message between each other, member such as (m_{1j}) sends a message to its master by citing the destination node of the message. When the node sends the request message to master it encrypts the message using its private key and sends it to the master. Suppose that a node p wants to send a message to a node q. At first p sends an encrypted message to it's master where it encrypts the message using its private key along with it's ID, receiver's ID and TS (e.g. is the time stamp or date and time when the authentication key is generated). Then the master decrypts the cipher text and identifies the node. The encryption and decryption process is as follows:

$$C_{pj} = E(K_{pr,pj}, (ID_{pj}||ID_{qj}||TS_{pj})) \tag{2}$$

$$M_{pj} = D(K_{pr,pj}, C_{pj}) \tag{3}$$

After decrypting the cipher text the master finds ids of the source (p) and destination (q) nodes. It computes a hash code for the nodes and authenticates them individually.

$$h_{pj} = H(ID_{qj}||K_{pr,pj}||CS_j) \tag{4}$$

$$h_{qj} = H(ID_{qj}||K_{pr,qj}||CS_j) \tag{5}$$

If $h_{pj} = K_{auth,pj}$ and $h_{qj} = K_{auth,qj}$, the nodes are authenticated, and they can communicate.

Figure 3 shows the overall flowchart of the authentication process.

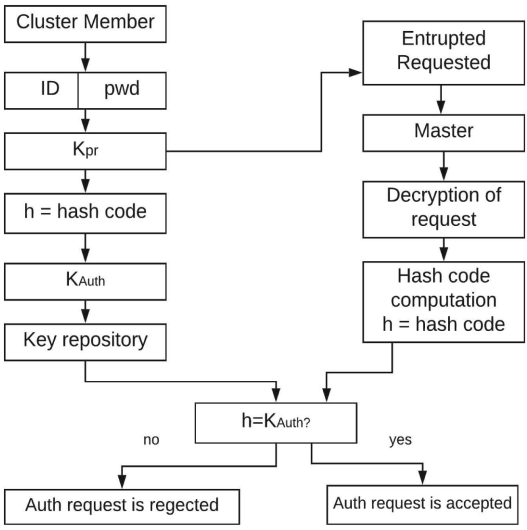


Fig. 3. Overall flowchart of the authentication process.

mmeybodi@aut.ac.ir

Without knowing the private key of authenticated node any malicious node or user cannot produce cipher text and send the message to the master. On the other hand, since CS_j is secret and stored in the master node, the service works for a trusted user only and it will not be possible or will be extremely difficult for a malicious node or user to authenticate itself and join a communication.

For key-encryption, the proposed I&AIoT service uses Elliptic Curve Encryption (ECE), which requires less key size compared to RSA cryptosystem and has fast processing power and less storage requirements [16]. As in IoT devices the required resource for public key primitives is much larger than that of symmetric key primitives [17], it was a traditional concern that any public/private encryption protocol would be computationally expensive. However, according to the authors of paper [18], computational complexity of public key cryptography is not anymore a blocking concern for IoT devices which natively support Elliptic Curve Cryptography (ECC). So, using ECE as the encryption technique makes our model both efficient and effective for both ample and limited resource lightweight IoT devices. We can also consider using new chip, which is designed by MIT (Massachusetts Institute of Technology) researchers based on elliptic-curve cryptosystem to perform public-key encryption that consumes only 1/400 as much power as software execution of the same protocol(s) would take [19].

Here inter cluster communications are made in the following way. All the cluster masters are the members of a supercluster (virtual). The supercluster has a special member called super master, which is a trusted administrator of the system. A supercluster with cluster masters is depicted in Fig. 4.

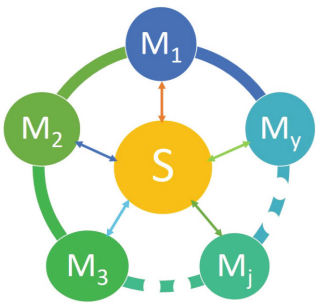


Fig. 4. A super cluster S with it's masters.

Here M_j is the master of the jth cluster and S is the super master. When cluster-cluster communication is needed the masters will be authenticated by the super master using its secret stored in it or his/her device in a similar way as the members of a cluster are authenticated. Any member of any cluster can send message to a member of another cluster through their masters. Here both masters will authenticate sender and receiver separately and allow a communication through the super master of the corresponding masters. Super master will authenticate each secure and insecure masters.

mmeybodi@aut.ac.ir

There are different use cases of our proposed authentication process. Let's consider a smart device which is a member m_{1j} in the cluster j and requesting for accessing another member m_{2j} in the same cluster. Here m_{1j} is the subject, m_{2j} is the object and M_j is the master of the corresponding cluster. Each and every member in cluster j must be registered under Master M_j. M_j would be familiar with the identification of all the devices or members in its cluster. In case of communication between cluster members we can find different use cases as following,

i. m_1 and m_2 both are new members (same cluster)
ii. m_1 is registered with M_j but not m_2 or vise versa (same cluster)
iii. m_1 and m_2 both are registered with M_j (same cluster)
iv. m_1 and m_2 are in different clusters.

3.1 Both Members are Unregistered

Subject m_{11} sends a request to be connected with object m_{21} to master M_1. M_1 verifies the registration status for both m_{11} and m_{21}. The sequence of requests for such process is shown in Fig. 5.

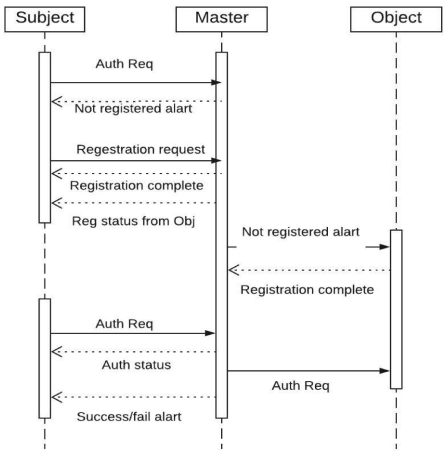


Fig. 5. Communication sequence when both members are unregistered.

In order to complete this, M_1 uses the id by deciphering the request from m_{11} and identifies the subject (m_{11}) and object (m_{21}) of the request. As both of the members are unregistered, M_1 sends a registration request to m_{11} first. In response, m_{11} submits the registration request with the required identification information and get registered under M_1. M_1 updates the member directory and sends the same request to m_{21}. m_{21} then gets registered under M_1 using the same process as m_{11}. If the registration process is failed for m_{11} or m_{21} the

mmeybodi@aut.ac.ir

authentication request is canceled immediately with the corresponding member of the failed registration. In this case, either m_{11} or m_{21} or both is listed as blacklisted member.

On the other hand, if the registration process is passed for both members, M_1 initiates authentication process for m_{11} and m_{21} immediately. If the authentication process is failed for either m_{11} or m_{21}, the request is canceled immediately, and a potential threat is logged in the threat log of M_1 with the corresponding member's information. Otherwise, if both the members are authenticated successfully, m_{11} and m_{21} are authenticated and connected for having secure conversation.

3.2 At Least One Member is Unregistered

There can be another case of communication where one member m_{11} is registered with cluster M_1 but another member m_{21} is not or vise versa.

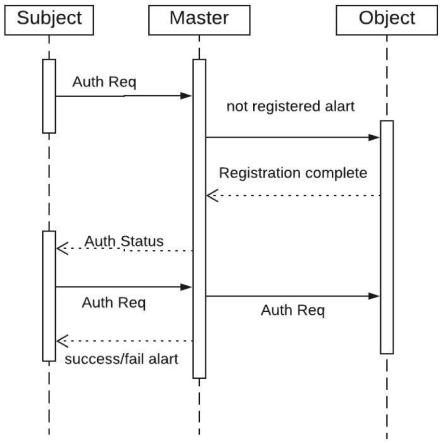


Fig. 6. Communication sequence when an object is a non-registered member.

Figure 6 shows a communication sequence where subject m_{11} is registered with master M_1 but object m_{21} is not. In this situation M_1 completes the registration process of m_{21} first, on a successful attempt. M_1 authenticates both m_{11} and textrmm$_{21}$ after the registration process of m_{21} and completes the authentication process for both m_{11} and m_{21}. If m_{21} is failed to provide required identification information the request is canceled immediately and m_{21} is listed as a blacklisted member in the directory of M_1 until m_{21} is re-registered successfully.

3.3 Both Members are Registered

When both m_{11} and m_{21} are registered under cluster M_1, M_1 initiates the authentication process immediately.

mmeybodi@aut.ac.ir

3.4 Both Members are in Different Clusters

For example, m_{11} is in cluster 1 along with Master M_1 and m_{21} is in cluster 2 along with master M_2. M_1 deciphers the message from subject m_{11} and identifies m_{11} using its cluster secret and the id of m_{11}. Once m_{11} is identified, M_1 computes the hash code using Eq. 4 and authenticates it. The next step for M_1 is to compute the hash code for m_{21} using Eq. 5 and send the result to super master S. S identifies both M_1 and M_2 and authenticates them by creating a corresponding hash code. If both master M_1 and M_2 are authenticated successfully, S sends an authentication request for m_{21} to M_2. M_2 identifies and authenticates m_{21} using the message as shown in Eq. 3 sent by S. Once m_{21} is identified and authenticated by M_2, M_2 notifies S and S sends a response to M_1 with successful authentication status. On receiving a successful response from S, M_1 initiates a connection request between m_{11} and m_{21} and submits the request to super master S. Figure 7 shows the sequence of requests among m_{11}, M_1, S, M_2, and m_{21}.

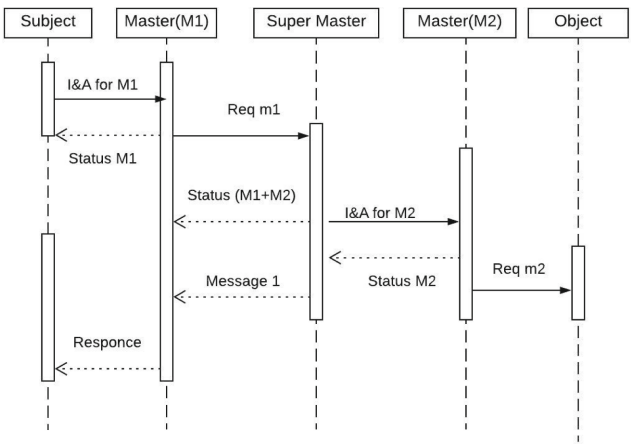


Fig. 7. Communication sequence between different cluster.

4 Performance Evaluation

In this section we explain that the proposed scheme presented above satisfies the major issues for ensuring identification and authentication of every user as well as device that are connected with IoT to establish a proper secure communication among them and also shows that this proposed scheme is the most effective one when taking into account some of the performance aspects.

mmeybodi@aut.ac.ir

4.1 Ensures Proper Identification and Authentication in a Significant Way

Our cluster based model includes id, password, private key generated by a cryptographic hash function such as SHA 256 in a proper way, described in the previous section, to ensure a secure communication between two members via master and super master. Any node or more specifically any malicious node or user cannot create ciphertext and send a message to a master or member as it does not have an authenticated private key to identify itself. Also, as the CS is secret and stored in only the corresponding master node, it will not be possible or will be extremely difficult to authenticate itself by any malicious node or user. Therefore, our proposed model assures proper secure communication not involving any malicious node which is not identified or authenticated.

4.2 Ensures Minimum Cost of Power and Memory Space

Our model is a cloud based model that provides service to every IoT device and performs the further operation within the cloud and this characteristic makes this system a light weight scheme which needs both minimum costs of power and memory space. Figure 8 shows a sample overall look of our model that describes how members communicate with masters and among themselves, masters communicate with super master and among themselves, super master communicates with another super master under a cloud service. Working under cloud service enables every involved device to experience convenient identification and authentication ensuring service within limited resource and power and eventually it broadens the service providing capacity of the system.

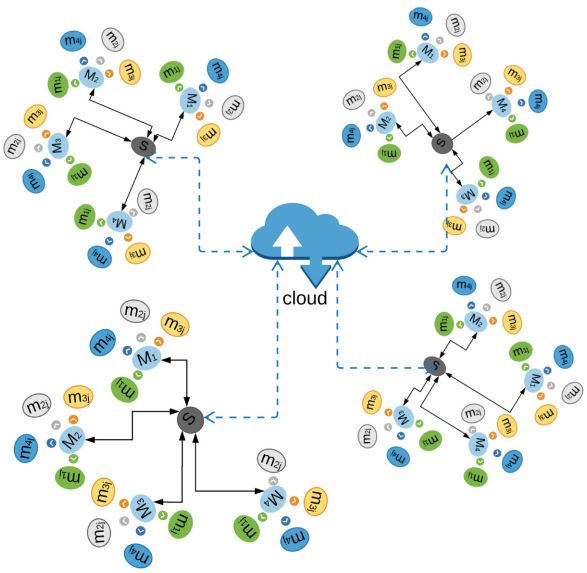


Fig. 8. Overall I&AIoT system.

mmeybodi@aut.ac.ir

4.3 Provides a Dynamic System for IoT

As every different kind of device has different configuration and also every device's data pattern is different, there still exists a need for a simple and dynamically configurable service that can be used by any device. So, our model is designed to be a dynamically configurable identification and authentication ensuring system for every kind of devices. To achieve this property, we have designed our proposed system using a really simple and formulated way where a cluster master identifies every user using their private key, generated by a cryptographic hash function using ID and password, and let communicate only the authenticated users.

4.4 Performs Immediate Threat Detection for Any Individual Cluster and Takes Proper Action

In the proposed model if the authentication process fails for any member (e.g. either m_1 or m_2) which is willing to communicate, the request is canceled immediately before letting it communicate with any other member and do any harm, and after that the potential threat is logged in the threat log of the corresponding master (e.g. M) which prohibits that particular malicious threat from trying to commute in future. This compelling feature of our I&AIoT model holds off some existing models like [12,14,15] in case of performance.

5 Conclusions

In this work, we have proposed a cluster based identification and authentication process for IoT platform. This cluster based proposed system uses cloud to compute some useful parameters for identification and authentication which makes it dynamically configurable and scalable architecture for both user and device. This architecture supports devices regardless of their types and resources by its simple and efficient service methods. It allows devices with a security protocol to exchange information through it's cluster-oriented identification and authentication checking based information exchange framework. The working process is also useful for cluster to cluster communication in an authentic way. As the main contribution, we have designed our proposed model in a cluster based significant and well organized way that establishes an effective safeguard to protect IoT users form any kind of threat caused by identification and authentication issues. Our model performs encryption, decryption and uses a hash function that ensures a proper security to the communication channel. In addition, our model uses cloud service that makes it a lightweight model with low cost and power consumption scheme. All these virtues together make the proposed architecture dynamically configurable for any complex IoT system such as smart cities, education and governance etc.

Here we have designed the I&AIoT system and defined useful components for it along with their need and operations as well as the working procedure. In future, we look forward to implementing this design and build a perfect I&AIoT software and make it usable for every IoT user as well as device.

mmeybodi@aut.ac.ir

References

1. Ashton, K.: That internet of things. https://www.rfidjournal.com/articles/view?4986
2. What is the IoT? Everything you need to know about the internet of things right now. https://www.zdnet.com/article/what-is-the-internet-of-things-everything-you-need-to-know-about-the-iot-right-now/. Accessed 4 Dec 2018
3. Ukil, A., Bandyopadhyay, S., Pal, A.: IoT-privacy: to be private or not to be private. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto (2014)
4. Gross, H., Holbl, M., Slamanig, D., Spreitzer, R.: "Privacy-Aware authentication in the Internet of Things," cryptology and network security, pp. 32–39. Springer (2015)
5. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., Zhao, W.: A survey on internet of things: architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J. **4**(5), 1125–1142 (2017)
6. Gazis, V.: A survey of standards for machine-to-machine and the Internet of Things. IEEE Commun. Surv. Tutor. **19**(1), 482–511 (2017)
7. Salman, O., et al.: Identity-based authentication scheme for the internet of things. In: 2016 IEEE Symposium on Computers and Communication (ISCC). IEEE (2016)
8. Shivraj, V. L., et al.: One time password authentication scheme based on elliptic curves for Internet of Things (IoT). In: 2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW). IEEE (2015)
9. Afifi, M.H., Zhou, L., Chakrabartty, S., Ren, J.: Dynamic authentication protocol using self-powered timers for passive Internet of Things. IEEE Internet Things J. **5**(4), 2927–2935 (2017)
10. Sungchul, L., Ju-Yeon, J., Yoohwan, K.: Method for secure RESTful web service. In: IEEE/ACIS, 14th International Conference on Computer and Information Science (ICIS 2015), Las Vegas-USA, pp. 77–81 (2015)
11. Liu, J., Xiao, Y., Chen, C.L.P.: Authentication and access control in the Internet of Things. In: IEEE 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW 2012), China, pp. 588–592 (2012)
12. Tewari, A., Gupta, B.B.: Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. J. Supercomput. **73**(3), 1085–1102 (2017)
13. Liu, W., et al.: The yoking-proof-based authentication protocol for cloud-assisted wearable devices. Pers. Ubiquit. Comput. **20**(3), 469–479 (2016)
14. Barreto, L., et al.: An authentication model for IoT clouds. In: 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE (2015)
15. Carrez, F., et al.: A reference architecture for federating IoT infrastructures supporting semantic interoperability. In: 2017 European Conference on Networks and Communications (EuCNC). IEEE (2017))
16. Luhach, A.K.: Analysis of lightweight cryptographic solutions for Internet of Things. Indian J. Sci. Technol. **9**(28) (2016)
17. Katagi, M., Moriai, S.: Lightweight cryptography for the Internet of Things. Sony Corporation (2008)
18. Sciancalepore, S., et al.: Public key authentication and key agreement in iot devices with minimal airtime consumption. IEEE Embed. Syst. Lett. **9**(1), 1–4 (2017)
19. Hardesty, L., MIT News Office.: Energy-efficient encryption for the internet of things, 12 February 2018. http://news.mit.edu/2018/energy-efficient-encryption-internet-of-things-0213

mmeybodi@aut.ac.ir

Forecasting of Customer Behavior Using Time Series Analysis

Hossein Abbasimehr[1(✉)] and Mostafa Shabani[2]

[1] Faculty of Information Technology and Computer Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran
abbasimehr@azaruniv.ac.ir
[2] IT Group, Department of Industrial Engineering, KN Toosi University of Technology, Tehran, Iran
mshabani@mail.kntu.ac.ir

Abstract. Forecasting future behavior of customers has significant importance in businesses. Consequently, data mining and prediction tools are increasingly utilized by firms to predict customer behavior and to devise effective marketing programs. When dealing with multiple time series data, we encounter with the problem that how to use those time series to forecast the behavior of all customers more accurately. In this study we proposed a methodology to create customer segments based on past data, create Segment-Wise forecasts and then discover the future behavior of each segment. The proposed methodology utilizes existing data mining and prediction tools including time series clustering and forecasting, but combines them in a unique way that results in higher level models in terms of accuracy than baseline model. The proposed methodology has substantial application in marketing for any firm in any domain where there is a need to forecast future behavior of different customer group in an effective manner.

Keywords: Time series analysis · ARIMA forecasting · Clustering · Customer behavior

1 Introduction

Data mining and machine learning tools and techniques have gained growing attention during recent years in all area applications such as marketing and business intelligence (BI) [1–4]. On the other hand, due to the advancements in information systems the huge amount of data is produced by businesses. In order to gain a deep understanding about their business and especially about their customers, many firms exploit BI tools [5, 6]. One of the area in which businesses uses BI techniques is customer behavior forecasting. Although customer behavior has various dimensions, modelling customer behavior in terms of their profitability is an attractive task that many firms attempt to accomplish it perfectly. It is important for a business to predict the future behavior of its customers to formulate proactive actions to respond to the threats and opportunities in an appropriate manner. Therefore, accuracy in forecasting of customer behavior is an important issue that a firm should deal with it.

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 188–201, 2020.
https://doi.org/10.1007/978-3-030-37309-2_15
mmeybodi@aut.ac.ir

In this study, we consider the attributes of the recency, frequency, and monetary (RFM) model [7] as customer behavior dimensions. To forecast customer activity in terms of RFM attribute values, the first requirement is to obtain appropriate data of past transactions. After obtaining the required data, data must be represented in a way to effectively tackle the problem at hand (e.g. forecasting). As we model data of customers as time series, so data analysis task will be faced with some challenges including the need for determining and specifying seasonality of data, noise and outlier management. The second requirement is that how to manage large population of customers and forecast the behavior and finally to construct a representative future time series that reflects the total behavior of customers. To deal with this requirements, we propose a methodology consisting of three approaches and implement them using data of a bank. The first approach which we called it as aggregate approach is a simple approach which firstly compute the mean of all customers' time series and uses it to forecast customers' behavior. The second approach that we named it as Segment-Wise forecasting divided into two sub-approaches including Segment-Wise-Aggregate (SWA) approach and Segment-Wise-Customer-Wise (SWCW) approach. The main characteristic of Segment-Wise methods is that they firstly perform clustering analysis on customer data which are represented in the form of time series data. Clustering step is accomplished by employing time series clustering techniques. An extensive set of experiments is conducted in order to find the best clustering results. Afterward, similar to baseline approach, the autoregressive integrated moving average model (ARIMA) [8, 9] method as a standard and widely-used method is used to time series forecasting. The accuracy of forecasting is evaluated using some accuracy measures (e.g. root mean square error). The results of this study on grocery guild indicates that the SWCW approach obtains a superior performance in terms of accuracy measures.

The reminder of the paper is organized as follows: Sect. 2 give some background on concepts and techniques utilized throughout of the paper. In Sect. 3, we describe the proposed methodology. Section 4 portrays the empirical study and the obtained results. In Sect. 5, we draw the conclusion.

2 Literature Review

2.1 RFM Model

RFM model is a popular model introduced by Hughes [7] which has been employed to measure customer life time value in various area of applications, for example, in retail banking [10, 11] in hygienic industry [12, 13] in retailing [14–18] in telecommunication [19, 20] in tourism [21]. Due to the significant importance of the monetary attribute (M) from banking viewpoint, in this study we interested in forecasting this attribute.

2.2 Time Series Clustering

A time series is defined as a sequence of data points ordered in time, typically in equal-length time intervals [22]. For example, suppose that a variable M is measured over n

mmeybodi@aut.ac.ir

time points then the time series M is denoted as $M = (m_1, m_2, \cdots m_{n-1}, m_n)$ where each m_i is the observation of M in time point i.

Time series clustering is considered as an especial kind of clustering [23, 24] which can be employed for various purposes including: discovering hidden patterns from data, exploratory analysis of data, sampling data and so on [26]. Given a set of time series data $D = \{M_1, M_2, \cdots, M_n\}$, time series clustering is the task of dividing of D into k partitions $C = \{c_1, c_2, \cdots, c_k\}$ such that similar time-series are grouped together based on a certain similarity measure. Then, c_i is denoted as a cluster where $D = \bigcup_{i=1}^{k} c_i$ and $c_i \cap c_j = \emptyset$ for $i \neq j$.

There are two key decisions in time series clustering including determining an appropriate dissimilarity measure between two time series data, and selecting a proper clustering algorithm.

Many dissimilarity measures have been proposed in the literature including Euclidean distance, dynamic time warping (DTW), temporal correlation coefficient (CORT), complexity-invariant distance measure (CID), discrete wavelet transform (DWT) and so on [27]. In the following subsection, we describe some of well-known dissimilarity criteria.

Regarding clustering algorithms, there have been many algorithms proposed which generally divided into four types comprising: partitioning-based, hierarchical, grid-based and density-based [23]. In this study, we use agglomerative hierarchical clustering algorithm for time series clustering as they have shown successful results in this context. Specifically, we employed the Ward method which is based on a sum-of-squares criterion. This method produces clusters that minimize within-cluster variance [28].

Dissimilarity Measures

To describe the following dissimilarity criteria, let us to define the two time series $X = (x_1, x_2, \cdots x_n)$ and $Y = (y_1, y_2, \cdots y_n)$ where n is the number of time-points.

Euclidean Distance

The Euclidean distance between the two time series X and Y is defined as [27]:

$$d_{L2}(X, Y) = \left(\sum_{t=1}^{n} (x_t - y_t)^2\right)^2 \tag{1}$$

Dynamic Time Warping

DTW [29] is a popular dissimilarity measure which is calculated based on finding the optimal alignment between two time series. The optimal path is searched using a dynamic programming approach [30, 31].

Considering two time series of X and Y, DTW distance can be described by equation

$$DTW(X, Y) = \min_{r \in M}\left(\sum_{m=1}^{M} |x_{im} - y_{jm}|\right) \tag{2}$$

Where the path element $r = (i, j)$ describes the association between two series. Since DTW is computed employing dynamic programming paradigm, this technique is expensive in computation [26].

mmeybodi@aut.ac.ir

Temporal Correlation Coefficient (CORT)
CORT takes into account both proximity on raw values and dissimilarity on temporal correlation behaviors when computing the similarity between two time series [27, 32]. It is defined as equation [27]

$$CORT(X,Y) = \frac{\sum_{t=1}^{n-1}(X_{t+1} - X_t)(Y_{t+1} - Y_t)}{\sqrt{\sum_{t=1}^{n-1}(X_{t+1} - X_t)^2}\sqrt{\sum_{t=1}^{n-1}(Y_{t+1} - Y_t)^2}} \tag{3}$$

Complexity-Invariant Distance Measure
CID which was developed by Batista, Keogh [33] computes the dissimilarity between two time series by estimating the complexity correction factor of the series [34]. A general CID measure is defined as [33]:

$$d_{CID}(X,Y) = CF(X,Y) \cdot d(X,Y) \tag{4}$$

Where $d(X,Y)$ corresponds to an existing distance measure, for example, Euclidean distance and CF is a complexity correction factor given by:

$$CF(X,Y) = \frac{max(CE(X), CE(Y))}{min(CE(X), CE(Y))}, \tag{5}$$

Where $CF(X)$ and $CF(Y)$ are complexity estimator of X and Y, respectively. For time series, $CF(X)$ can be computed as follows:

$$\mathrm{CE(X)} = \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}-1}(\mathrm{x_i} - \mathrm{x_{i+1}})^2} \tag{6}$$

Discrete Wavelet Transform
Discrete wavelet transform (DWT) is another popular technique employed to measure similarity between time series [27]. DWT substitutes the original time series by their wavelet approximation coefficient in a proper scale, and then measure dissimilarity based on the wavelet approximations [27]. More information on wavelet methods in the context of time series clustering can be seen in Percival and Walden [35].

2.3 Time Series Forecasting

ARIMA
ARIMA modeling [8] is one of the popular and widely-used techniques to time series forecasting. For modeling, the ARIMA can represent various modeling types of stochastic seasonal and nonseasonal time series such as pure autoregressive (AR), pure moving average (MA) and mixed AR and MA models [36].

mmeybodi@aut.ac.ir

The multiplicative seasonal ARIMA model, represented as ARMIA $(p,d,q) \times (P,Q,D)_m$ has the following form [9]:

$$\phi_p(B)\Phi_P(B^m)(1-B)^d(1-B^m)^D y_t = c + \theta_q(B)\Theta_Q(B^m)\varepsilon_t \tag{7}$$

Where

$$\phi_p(B) = 1 - \phi_1 B - \cdots - \phi_p B^p,\ \Phi_P(B^m) = 1 - \Phi_1 B - \cdots - \Phi_P B^P \tag{8}$$

$$\theta_q(B) = 1 + \theta_1 B + \cdots + \theta_q B^q,\ \Theta_Q(B^m) = 1 + \Theta_1 B + \cdots + \Theta_Q B^Q \tag{9}$$

And m is the seasonality frequency,B is the backward shift operator,d is the degree of ordinary differencing, and D is the degree of seasonal differencing, $\phi_p(B)$ and $\theta_q(B)$ are the regular autoregressive and moving average polynomials of orders p and q, respectively, $\phi_p(B)$ and $\Theta_Q(B^m)$ are the seasonal autoregressive and moving average polynomials of orders P and Q, respectively, $c = \mu(1 - \phi_1 - \cdots - \phi_p)(1 - \Phi_1 - \cdots - \Phi_p)$ where μ is the mean of $(1-B)^d(1-B^m)^D y_t$ process and ε_t is zero mean Gaussian white noise process with variance σ^2. The roots of the polynomials.

3 Proposed Methodology

The proposed methodology for customer behavior forecasting is portrayed in Fig. 1. The methodology is divided into three main steps including Preprocessing, Modelling and Evaluation. In the following, we describe each step briefly.

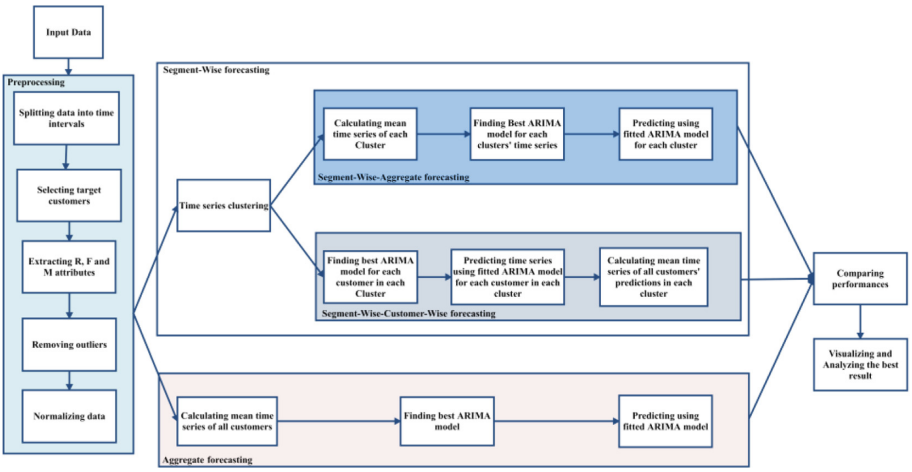


Fig. 1. The steps of proposed methodology

mmeybodi@aut.ac.ir

3.1 Input Data

The input of this methodology is the customers' past purchases data

3.2 Preprocessing

In this step, cleaning and transforming data into RFM model attributes are performed using the following steps.

Splitting Data into Proper Time Intervals
As the time series data is used in this model. The data must be divided into time intervals. So, the customers' data are aggregated at each time points.

Selecting Target Customers
In this step, based on attributes for each customers and the resulted data from previous step, the customers who have value in all time points are filtered.

Extracting R, F and M Attributes
The proposed methodology is based on RFM model, so the data for a time point must be transformed into R, F and M attributes of RFM model. The R attribute is the days between the date of last purchase and the date of end of the time point. F attribute is the frequency of purchases in a time point. M attribute is the total amount of purchases in a time point.

Removing Outliers
The incorrect data or data with anomaly values are removed. In this step each attribute of RFM model for each time point are evaluated under an anomaly detection algorithm [23] and the outliers are removed.

Normalizing Data
Each time point is analyzed independently so the data for each time point normalized separately. The Min-Max normalization algorithm is used in this model.

3.3 Modelling

In this step, we proposed three approaches for time series forecasting that are as follows:

Aggregate Forecasting
Aggregate forecasting is the baseline approach of forecasting which is based on aggregating all customers' RFM model attributes. The steps in this phase are as following:

Calculating Mean Time Series of all Customers
In this step for each attribute of RFM model the mean value of all customers is calculated. These values are used for time series prediction in the next steps.

Finding the Best ARIMA Model
Using the mean time series of all customers, the best ARIMA model is built.

mmeybodi@aut.ac.ir

Predicting Using the Fitted ARIMA Model
In this step, the fitted model is used to predict future values. The performance of the model is evaluated using evaluation measures.

Segment-Wise Forecasting
In this subsection, we describe the Segment-Wise forecasting methods.

Time Series Clustering
This phase is based on the idea that the time series forecasting of customer segments with the same behaviors over time can be more accurate than forecasting of all customers without any behavioral segmentation. For this purpose, in this step the best time series similarity measures are selected and hierarchical clustering with the best linkage methods is implemented. The outcome of this step is the customer segments with the same behavior over time.

Segment-Wise-Aggregate (SWA) Forecasting
In this strategy of customer time series forecasting, mean values of RFM model attributes for each cluster are calculated. Forecasting model based on ARIMA model for each cluster is built and prediction based on constructed model is generated.

Calculating Mean Time Series of Each Cluster
Using the resulted segments of customers from the clustering step, the mean time series for each attributes of RFM model are calculated.

Finding the Best ARIMA Model for Each Customer Segment
For each segment, the best ARIMA model is built.

Predicting Using Fitted ARIMA Model for Each Cluster
Time series prediction using fitted model is generated in this step and evaluating parameters generated for the next phase.

Segment-Wise-Customer-Wise (SWCW) Forecasting
This strategy of forecasting is based on forecasting the future values for each customer separately. Calculate mean time series of all customers' predictions. The steps in this strategy are as following:

Finding the Best ARIMA Model for Each Customer in Each Cluster
For each customer, the best ARIMA model is obtained.

Predicting Time Series Using Fitted ARIMA Model for Each Customer in Each Cluster
By using the fitted models for each customer in each cluster the future values are predicted.

Calculating Mean Time Series of all Customers' Prediction in Each Cluster
As all customers' prediction time series for each cluster are generated, mean value of all prediction in each cluster is used as the predicted time series for each cluster.

mmeybodi@aut.ac.ir

3.4 Evaluation

To test the performance of built models, we utilized the root mean square error (RMSE), and symmetric mean absolute percentage error (SMAPE) [37] to measure the performance of the ARIMA models.

RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2} \tag{10}$$

Where y_t and $\hat{y}_t$ are the actual and forecast values of the series in time point t respectively.

In addition, SMAPE is represented by:

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|\hat{y}_t - y_t|}{\frac{|\hat{y}_t| + |y_t|}{2}} \tag{11}$$

Where y_t and $\hat{y}_t$ are the actual and forecast values of the series in time point t respectively.

4 Empirical Study and Analysis

4.1 Input Data

In this study we used transactions of POS customers, total transactions are 1,200,000. A sample of data with the features are shown in Table 1. Terminal ID is the ID of POS device that a customer used; Transaction Date field indicates the date of a transaction; Transaction Amount is the value of a transaction in IRI currency; and finally Terminal Guild field shows the guild in which each customer belongs to it.

Table 1. A sample of data for illustration of the input data

Terminal ID	Transaction Date	Transaction Amount (IRI)	Terminal Guild
3128803	01052018	344002	11
3129948	01052018	982000	11
3136664	01052018	2700000	3
3143083	01052018	542201	8
3166247	01052018	1200000	1
3166657	01052018	1800000	11

Since the ultimate goal of any firm often is reaching the desired profitability, hence, we only use the Monetary attribute as a representation of customer behavior. Therefore, in this study, we consider the problem of the prediction of the future behavior of customers in terms of Monetary.

mmeybodi@aut.ac.ir

4.2 Preprocessing

Splitting Data into Proper Time Intervals
We divided our daily data to weekly data to make it more manageable. As the gathered data is for 11 months, the resulted data consists of 44 time points.

Selecting Target Customers
In our experiment, we concentrated on analyzing active customers which are defined as customers who have transactions in all time points. Total active customers are 123000 customers. As in our data we have guild field, we chose a specific guild for analysis.

Extracting R, F and M Attributes
As the model is based on the RFM model, RFM model attributes were derived from the data.

Removing Outliers
To reduce the effect of outliers, we carried out outlier detection using standard deviation.

Normalizing Attributes
The Min-Max normalization algorithm [23] was used in this step.

4.3 Modelling

In this step, for each approach, we used auto.arima function in the forecast package for R [38] to find the best ARIMA model.

Aggregate Forecasting
Based on the definition of this approach in Sect. 3, this is the baseline method which doesn't consider the clustering step. It works based on forecasting the mean time series of all customers using ARIMA model. The results and evaluation of this strategy is presented in next subsection.

Segment-Wise Forecasting
As described in proposed model section, to implement this approach, time series clustering was accomplished and the outcome results used for forecasting. The best time series clustering based on the silhouette validity index [39] as can be seen in Table 2 is clustering with CID and k = 4.

Table 2. The silhouette index for each combination of cluster numbers (K) and distance measures

Distance measure	K = 4	K = 5	K = 6	K = 7	K = 8
Euclidean	0.13	0.13	0.13	0.14	0.14
CORT	0.17	0.17	0.16	0.16	0.17
DTW	0.21	0.21	0.22	0.15	0.15
CID	**0.4**	0.28	0.28	0.24	0.28
DWT	0.37	0.38	0.38	0.39	0.39

mmeybodi@aut.ac.ir

Table 3. Size of the obtained clusters

Cluster number	Size
Cluster 1	99
Cluster 2	88
Cluster 3	40
Cluster 4	30

The population of each customer segment using CID algorithm with 4 clusters is illustrated in Table 3.

Our analysis is concentrated on M attribute of RFM model. For the SWA forecasting, the mean value of M attribute for each cluster is calculated and ARIMA model built based on that time series. The forecasting for each cluster conducted using proper fitted model.

In the SWCW forecasting, time series forecasting for each customer using ARIMA model is performed and the mean value of all forecast data is the forecast time series for each cluster.

The results and evaluation of these strategies are presented in next subsection.

4.4 Evaluation

In the following, we have given the results of the three approaches in terms RMSE and SMAPE (Table 4). As seen from Table 4, the SWCW approach outperforms other methods. Therefore, in the following we compare the performance of the two approach that are categorized as Segment –Wise approach.

Table 4. Performance of the three forecasting methods in terms of RMSE and SMAPE

Forecasting method	RMSE	SMAPE
Aggregate Forecasting	0.045	0.59
Segment-Wise-Customer-Wise(SWCW)	**0.0344**	**0.3818**
Segment-Wise-Aggregate(SWA)	0.0468	0.8584

Table 5 summarized the results of forecasting using Segment-Wise methods. As indicated in Table 5, the SWCW forecasting approach outperforms the SWA in terms of RMSE and SMAPE. In addition, for better comparison of the results, the results of forecasting of 8 time points (test split) for the Segment-Wise approaches are illustrated in Figs. 2, 3, 4 and 5. These figures show the actual data value (dashed black line), the value predicted by the SWA forecasting method (red color) and the value predicted by SWCW method (green color). As seen from Figs. 2, 3, 4 and 5, the SWCW approach has a higher forecasting power than the SWA method.

mmeybodi@aut.ac.ir

Table 5. Results of forecasting using SWA and SWCW methods

Segment	Segment-Wise-Customer-Wise		Segment-Wise-Aggregate	
	RMSE	SMAPE	RMSE	SMAPE
Segment 1	0.017	0.388	0.023	0.8551
Segment 2	0.036	0.398	0.053	0.995
Segment 3	0.049	0.29	0.052	0.265
Segment 4	0.068	0.436	0.1	1.26
Micro-average	**0.0344**	*0.3818*	**0.0468**	*0.8584*

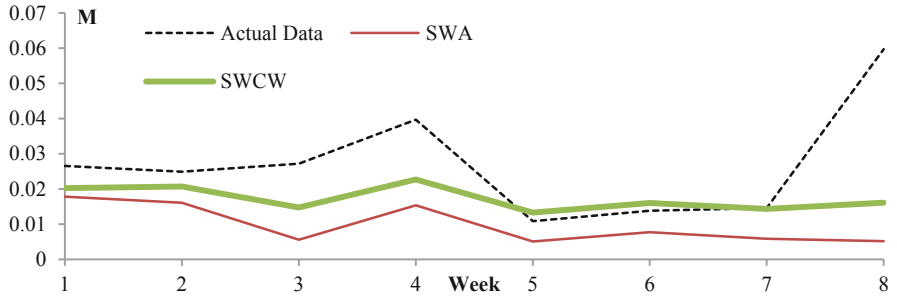


Fig. 2. Forecasting segment 1 future values using SWA and SWCW approaches

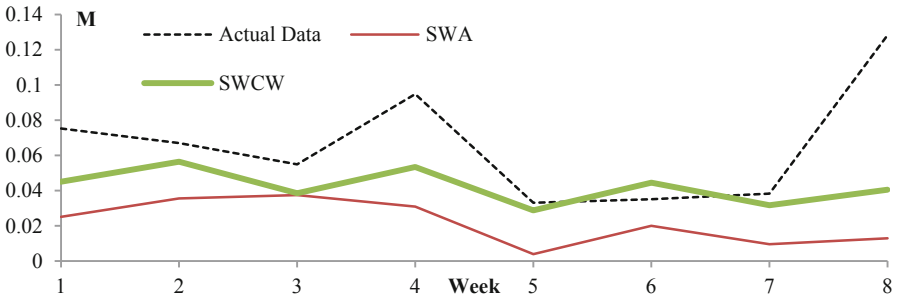


Fig. 3. Forecasting segment 2 future values using SWA and SWCW approaches

The results of this study indicated that SWCW method outperformed the SWA method. It is worth to note, that the results of this research are limited to the available data. Therefore, the results may not generalizable to other time series data. However, the proposed methodology can be employed in other domains to analyze behavior of customers.

mmeybodi@aut.ac.ir



Fig. 4. Forecasting segment 3 future values using SWA and SWCW approaches

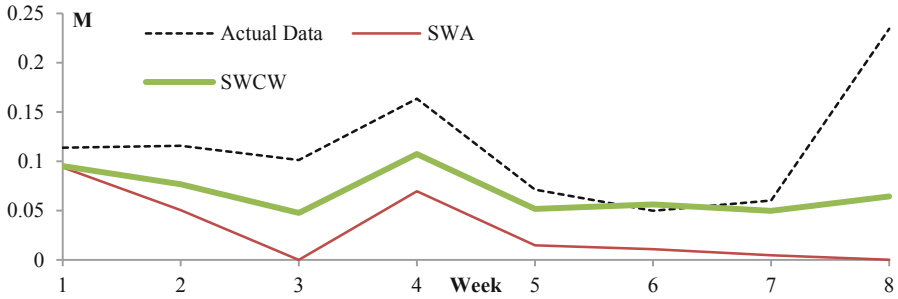


Fig. 5. Forecasting segment 4 future values using SWA and SWCW approaches

5 Conclusion

Forecasting future behavior of customers is one of the main purposes of almost any firm in any domain. In this study, we proposed a combined methodology to forecast customer behavior. This methodology combines the state-of-the-art data mining and time series analysis techniques including time series clustering along with time series forecasting using ARIMA model. the methodology describes the essential steps of fore casting including preprocessing, modelling and evaluation. We considered RFM attributes as customer behavior dimensions. In order to demonstrate the application of the proposed methodology, we have carried out a case study on data of a bank in Iran. Results of case study indicated that Segment-Wise-Customer-Wise (SWCW) method outperforms the other methods in terms of accuracy measures including RMSE and SMAPE. This method, can be able to predict future behavior of different segments of customers effectively. The proposed combined method can be utilized in other domains to predict customers' future behavior.

mmeybodi@aut.ac.ir

References

1. Kumar, V., Reinartz, W.: Customer Relationship Management: Concept, Strategy, and Tools. Springer, Heidelberg (2018)
2. Chiang, W.-Y.: Applying data mining for online CRM marketing strategy: an empirical case of coffee shop industry in Taiwan. Br. Food J. **120**(3), 665–675 (2018)
3. Yildirim, P., Birant, D., Alpyildiz, T.: Data mining and machine learning in textile industry. Wiley Interdisc. Rev.: Data Min. Knowl. Discov. **8**(1), e1228 (2018)
4. Lessmann, S., et al.: Targeting customers for profit: an ensemble learning framework to support marketing decision making (2018)
5. Duan, Y., Cao, G., Edwards, J.S.: Understanding the impact of business analytics on innovation. Eur. J. Oper. Res. **281**, 673–686 (2018)
6. Grover, V., et al.: Creating strategic business value from big data analytics: a research framework. J. Manag. Inf. Syst. **35**(2), 388–423 (2018)
7. Hughes, A.: Strategic Database Marketing: The Masterplan for Starting and Managing a Profitable, Customer-Based Marketing Program, 4th edn. McGraw-Hill Companies, Incorporated, USA (2011)
8. Box, G.E., et al.: Time Series Analysis: Forecasting and Control. Wiley, Hoboken (2015)
9. Brockwell, P.J., Davis, R.A., Calder, M.V.: Introduction to Time Series and Forecasting. Springer, Heidelberg (2002)
10. Khajvand, M., Tarokh, M.J.: Estimating customer future value of different customer segments based on adapted RFM model in retail banking context. Proc. Comput. Sci. **3**, 1327–1332 (2011)
11. Hosseini, M., Shabani, M.: New approach to customer segmentation based on changes in customer value. J. Mark. Anal. **3**(3), 110–121 (2015)
12. Parvaneh, A., Abbasimehr, H., Tarokh, M.J.: Integrating AHP and data mining for effective retailer segmentation based on retailer lifetime value. J. Optim. Ind. Eng. **5**(11), 25–31 (2012)
13. Parvaneh, A., Tarokh, M., Abbasimehr, H.: Combining data mining and group decision making in retailer segmentation based on LRFMP variables. Int. J. Ind. Eng. Prod. Res. **25** (3), 197–206 (2014)
14. Hu, Y.-H., Yeh, T.-W.: Discovering valuable frequent patterns based on RFM analysis without customer identification information. Knowl.-Based Syst. **61**, 76–88 (2014)
15. You, Z., et al.: A decision-making framework for precision marketing. Expert Syst. Appl. **42** (7), 3357–3367 (2015)
16. Abirami, M., Pattabiraman, V.: Data mining approach for intelligent customer behavior analysis for a retail store, pp. 283–291. Springer, Cham (2016)
17. Serhat, P., Altan, K., Erhan, E.P.: LRFMP model for customer segmentation in the grocery retail industry: a case study. Mark. Intell. Plann. **35**(4), 544–559 (2017)
18. Doğan, O., Ayçin, E., Bulut, Z.A.: Customer segmentation by using RFM model and clustering methods: a case study in retail industry. Int. J. Contemp. Econ. Adm. Sci. **8**(1), 1–19 (2018)
19. Akhondzadeh-Noughabi, E., Albadvi, A.: Mining the dominant patterns of customer shifts between segments by using top-k and distinguishing sequential rules. Manag. Decis. **53**(9), 1976–2003 (2015)
20. Song, M., et al.: Statistics-based CRM approach via time series segmenting RFM on large scale data. Knowl.-Based Syst. **132**, 21–29 (2017)
21. Dursun, A., Caber, M.: Using data mining techniques for profiling profitable hotel customers: an application of RFM analysis. Tour. Manag. Perspect. **18**, 153–160 (2016)

mmeybodi@aut.ac.ir

22. Le, D.D., Gross, G., Berizzi, A.: Probabilistic modeling of multisite wind farm production for scenario-based applications. IEEE Trans. Sustain. Energy **6**(3), 748–758 (2015)
23. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques: Concepts and Techniques. Elsevier Science, Amsterdam (2011)
24. Witten, I.H., et al.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2016)
25. Tan, P.-N.: Introduction to Data Mining. Pearson Education India (2006)
26. Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.Y.: Time-series clustering – a decade review. Inf. Syst. **53**, 16–38 (2015)
27. Montero, P., Vilar, J.A.: TSclust: an R package for time series clustering. J. Stat. Softw. **62** (1), 1–43 (2014)
28. Murtagh, F., Legendre, P.: Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? J. Classif. **31**(3), 274–295 (2014)
29. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Sig. Process. **26**(1), 43–49 (1978)
30. Anantasech, P., Ratanamahatana, C.A.: Enhanced weighted dynamic time warping for time series classification. In: Third International Congress on Information and Communication Technology, pp. 655–664. Springer (2019)
31. Mueen, A., et al.: Speeding up dynamic time warping distance for sparse time series data. Knowl. Inf. Syst. **54**(1), 237–263 (2018)
32. Chouakria, A.D., Nagabhushan, P.N.: Adaptive dissimilarity index for measuring time series proximity. Adv. Data Anal. Classif. **1**(1), 5–21 (2007)
33. Batista, G.E., et al.: CID: an efficient complexity-invariant distance for time series. Data Min. Knowl. Discov. **28**(3), 634–669 (2014)
34. Cen, Z., Wang, J.: Forecasting neural network model with novel CID learning rate and EEMD algorithms on energy market. Neurocomputing. **317**, 168–178 (2018)
35. Percival, D.B., Walden, A.T.: Wavelet Methods for Time Series Analysis. Cambridge University Press, Cambridge (2006)
36. Ramos, P., Santos, N., Rebelo, R.: Performance of state space and ARIMA models for consumer retail sales forecasting. Robot. Comput.-Integr. Manuf. **34**, 151–163 (2015)
37. Martínez, F., et al.: Dealing with seasonality by narrowing the training set in time series forecasting with kNN. Expert Syst. Appl. **103**, 38–48 (2018)
38. Hyndman, R., et al.: Forecast: forecasting functions for time series and linear models. In: R Package Version 8.4 (2018)
39. Desgraupes, B.: Clustering indices, vol. 1, p. 34. University of Paris Ouest-Lab Modal'X (2013)

mmeybodi@aut.ac.ir



Correlation Analysis of Applications' Features: A Case Study on Google Play

A. Mohammad Ebrahimi, M. Saber Gholami, Saeedeh Momtazi(✉), M. R. Meybodi, and A. Abdollahzadeh Barforoush

Department of Computer Engineering,
Amirkabir University of Tehran, Tehran, Iran
{amirebrahimi, sabergh, momtazi, mmeybodi, ahmadaku}@aut.ac.ir

Abstract. The presence of smartphones and their daily usages have changed several aspects of modern life. Android and IOS devices are widely used these days by the public. Besides, enormous number of mobile applications have been developed for the users. Google launched an online market which is known as Google Play for offering applications to end users as well as managing them in an integrated environment. Applications have many features that developers should clarify while they are uploading apps. These features have potential correlations which studying them could be useful in several tasks such as detecting malicious or miscategorized apps. Motivated by this, the purpose of this paper is to study these correlations through Machine Learning (ML) techniques. We apply various ML classification algorithms to distinguish these relations among key features of applications. Additionally, we perform many examinations to observe the relations between the size of the feature vector and the accuracy of the mentioned algorithms. Furthermore, we compare the algorithms to find the best choices for each part of our experiments. The results of our evaluation are promising. Also, in the majority of cases there are strong correlations between features.

Keywords: Mobile devices · Machine learning · Natural Language Processing · Data analysis · Feature engineering

1 Introduction

In recent years usage of smartphones has been increased. They evolved from simple devices to smart ones that enable users to do various tasks like emailing, navigation, communicating with others, browsing on the internet, taking photos, gaming, etc. These tasks can be done through applications. Furthermore, smartphones need an operating system (O.S) in order to manage all the mentioned tasks. There are several O. S for these devices i.e. IOS, Android, Blackberry and Symbian. Recent advances in the context of applications lead to tight competition among developers to build brand new products. Consequently, many applications have been developed, and they demand huge markets to be organized. For satisfying this requirement, several online markets have been founded. Apple store was the first online market that implied this demand. Afterward, Google introduced "Google Play" for Android users.

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 202–216, 2020.
https://doi.org/10.1007/978-3-030-37309-2_16

mmeybodi@aut.ac.ir

At March 2017 Google announced that Android has more than 2 billion monthly active devices. Google also has a lot of other services such as Gmail, YouTube, Google Maps, Chrome and Google Play with over one billion active users in a month [1]. Google Play is an online app store which launched in 2008. At the moment, it has more than 3.6 million available apps in diverse categories [2]. These magnificent number of applications provide a massive amount of worthwhile information, which revolutionize research in many data science areas i.e. security analysis, store ecosystem, release engineering, review analysis, API usage, prediction and feature analysis [3].

Feature Analysis is broken down into many subcategories like "Classification", "Clustering", "Lifecycles", "Recommendation" and "Verification" [3]. From Classification perspective one of the most critical concerns is selecting a set of appropriate features. Papers in this field extract features from multiple sources such as an application's information in Google Play page or binary files of a specific app, with the aim of feeding them into a classifier to distinguish categories and find miscategorized applications [4–8]. Additionally, checking app security, detecting malicious behaviors, and identifying usage of sensitive information (e.g. 'location' and 'contact') have been studied in these scopes [9–12].

There are several different attributes that can be used for feature selection. One way is selecting features from an application's page in Google Play. It has various informative data for each application i.e. Permissions, Description, User reviews, Rate and so on. These features can be used along with other features from other sources for training a classifier to predict a target. In comparison, if selected features contain raw text, converting text into understandable data for a classifier is a critical concern. "Feature Selection" algorithms are a promising solution for this matter. Also, there are too many machine learning classification methods that can be applied for predicting. So picking the suitable algorithms can affect classification performance.

Motivated by these facts, we launched a study to investigate the mentioned topics. Our ultimate objective is studying the main features from google play that can be predicted by other features and finding the correlation between features when predicting the target feature. The contribution of this work is three-fold:

- We gathered approximately 7311 applications from Google Play pages[1].
- We employed eight classification techniques to categorize and predict multiple targets with the purpose of finding the best method for predicting each target.
- We compared the performance of the classifiers for every possible combination of features in order to study the correlation between them.

The remainder of this paper is structured as follows: We start in Sect. 2 with a survey of previous relevant studies. Section 3 explores Google Play structure and features that are available online. Section 4 describes the methods we used and also defines different kind of features. Section 5 discusses feature engineering phase and preprocessing. In Sect. 6 we present experimental setup and evaluation results. We also discuss some practical usage of our findings. Finally, Sect. 7 discusses the results.

[1] Available online at: https://github.com/sabergh/Google_Play_Applications.

mmeybodi@aut.ac.ir

2 Related Works

Recent studies in app stores could be divided into seven categories i.e. security, store ecosystem, size and effort prediction, API usage, feature analysis, release engineering and reviews [3]. Two of these fields are related to our work which are "security" and "feature analysis". Researches on security domain try to identify potentially harmful behaviors which are malware detection and inappropriate usage of permissions. On the other hand, papers in feature analysis aim at extracting features out of different sources and use them for classification, recommendation systems, clustering, etc. Papers in these two categories will be discussed below.

Security Varma et al. [12] attempted to detect malicious apps based on the requested permissions. They performed and compared five machine learning algorithms to predict suspicious applications. For training the classifiers, they extracted permissions of applications out of the manifest file and used them as classifiers' features. Gorla et al. [13] proposed CHABADA framework which tries to distinguish trustable apps from dangerous ones by making a contrast between app's description and the usage of API. They performed an LDA topic modeling to find the number of appropriate categories and used k-means to categorize applications. In comparison, Ma et al. [14] performed a semi-supervised learning method to the same problem and achieved higher performance than CHABADA. Shabtai et al. [15] aimed to apply machine learning techniques on an application byte-code for classifying applications into two categories i.e. games and tools. They also suggest that this successful categorization could be used for detecting suspicious behaviors of an application.

Feature Analysis Liu et al. [9] aimed to decide whether an application is suitable for children or not with SVM classifier. They used a variety of features like app category, content rating, title, description and its readability, picture, and texts on it. After that, they generated a list of suitable apps for kids. Olabenjo [8] suggested an appropriate category for new applications. He mined more than 1 million applications and reduced this number to approximately 10000 by removing all applications that have been not developed by top-developers. Then used five features of each application contains app name, content rating, description, whether the application is free or not and whether it has in-app-purchase or not. After that, he performed Bernoulli and Multinomial Naïve Bayes. Berardi et al. [6] focused on presenting an automatic system for suggesting the category of an application which is based on the user's demands. They crawled approximately 6000 applications and extracted the main features of each. Then performed SVM classifier to predict the category of applications. They reached an accuracy of 0.89 which is highly dependent on the imbalance rate of data. In other words, 84.6% of mined applications were in the same category. In 2017 Surian et al. [5] introduced FRAC+: a framework for app categorization which aims to suggest an appropriate category for new applications and also detect the miscategorized ones. The framework consists of two main sections: (i) calculate the optimal number of categories (ii) running the topic model with the calculated number.

However, based on the above discussion, in this paper, we studied the classification of Google Play applications with various learning models and with every possible permutation of features which highly differs from prior studies.

mmeybodi@aut.ac.ir

3 Google Play and Dataset

Google Play launched in March 2012 and is an online market that Android developers use to offer their applications. People use applications to satisfy their needs like massaging, photography, playing, emailing, communicating with others on social media, etc. Each application in Google Play has various features. In this section, we discuss these features and clarify their distribution and scaling in our data set.

Every application has many attributes in Google Play. These attributes divided into two main types: (i) attributes that are available on application's page and filled by the developer like name, developer's name, suggested categories, number of downloads, user ratings, description, reviews, last update, size, current version, Android version, content rating and permissions list. (ii) Additionally, there is another type of features that could be extracted from applications byte-code or manifest file. In this paper, we concentrate on the first type.

We crawled 7311 applications from google play. The number is reduced to 6668 by removing non-English apps. The distributions of these applications which are divided into 48 classes are shown in Table 1. To reduce the number of classes, we merged similar categories based on their functionality. In Table 1 there is a number in parentheses in front of each category that illustrates the mapping to the new categories.

Table 1. Distribution of crawled applications categories

Category	Frq	Category	Frq	Category	Frq
Action (1)	252	Education (3)	434	Food & Drink (7)	33
Adventure (1)	203	Books & References (3)	75	Health & Fitness (7)	198
Racing (1)	203	News & Magazines (3)	112	Lifestyle (7)	90
Role playing (1)	172	Auto & Vehicles (4)	17	Medical (7)	7
Simulation (1)	262	House & Home (4)	28	Parenting (7)	46
Trivia (1)	76	Maps & Navigation (4)	59	Music & Audio (8)	178
Arcade (2)	319	Shopping (4)	71	Photography (8)	187
Board (2)	55	Travel & Local (4)	79	Vide players-Editors(8)	65
Card (2)	54	Weather (4)	63	Personalization (9)	236
Casino (2)	51	Comics (5)	32	Dating (10)	5
Casual (2)	463	Entertainment (5)	268	Communicating (10)	134
Educational (2)	400	Libraries & Demo (5)	16	Social (10)	115
Music (2)	80	Sports (5)	241	Art & Design (11)	66
Puzzle (2)	364	Business (6)	64	Events (11)	14
Strategy (2)	169	Finance (6)	76	Productivity (11)	194
Word (2)	90	Beauty (7)	10	Tools (11)	242

mmeybodi@aut.ac.ir

Table 2. Distribution of new assigned categories

Category	Frq	Category	Frq	Category	Frq	Category	Frq
1	1168	2	2045	3	621	4	317
5	557	6	140	7	384	8	430
9	236	10	254	11	516		

Overall, we suggested 11 categories that could be found in Table 2 with distributions. As the data was fine-grain, we faced another problem for values and scaling of features such as "rating", "size" and "number of downloads". For example, users can rate an app between 0 to 5 stars so the average rating of an app could be a float number like $0.1, 0.2, \ldots, 4.9, 5.0$. We merged these to 0, +1, +2, +3 and +4. Since the size of applications is in Megabyte, we change them to a coarse-grained scaling by dividing them into categories like 0–10 Mb, 10–20 Mb, etc. Additionally, as 39% of applications do not have the size or the developer mentions: "varies with devices", we put a label "NaN" for them. Furthermore, we used the same solution to tackle with "number of downloads". Therefore all applications that have a close number of installations merged into the same category.

4 Classification Algorithms

In this section, we explore several machine learning algorithms which are used in our experiments. Machine learning is a field of research in artificial intelligence that uses statistical techniques to allow computer systems to learn from data and getting them to act without being explicitly programmed [17].

Generally, machine learning algorithms can be divided into categories based on their purpose or type of training data. From the training data perspective, there are three approaches: supervised learning, unsupervised learning, and semi-supervised learning. In supervised learning, each of the training examples in training dataset must be labeled, then the algorithm analyzes the training data and produces a model which can be used to label unseen examples [18]. Unsupervised algorithms learn from training data that has not been labeled, so the learning process is based on the similarity between the training examples [19]. Semi-supervised learning falls between supervised learning and unsupervised learning because it uses a mixture of labeled and unlabeled data. In comparison with supervised learning, this approach helps to reduce the cost and effort of labeling data. Also, the small proportion of labeled data used in this approach improves the classification accuracy compared to unsupervised learning [20].

In this paper, we used supervised learning for two reasons: first, our experiments are based on classifying different targets. Additionally, the selected data are properly labeled by developers, therefore we do not need to put any effort into annotation. Following the above discussion, the supervised classification algorithms, which are used in our experiments, will be introduced below.

Naïve Bayes (NBs) algorithms are a set of common supervised learning algorithms based on applying Bayes' theorem. One of the most important principles of NBs is

mmeybodi@aut.ac.ir

"naïve" assumption which considers every feature independent of others [21]. Despite being simple, they work quite well in real-world scenarios. They demand much less labeled data comparing to other learning algorithms. Furthermore, concerning runtime, they can be extremely fast compared to more sophisticated methods. In this paper, we experimented different versions of Naïve Bayes algorithm i.e. Bernoulli Naïve Bayes (BNB), Gaussian Naïve Bayes (GNB) and Multinomial Naïve Bayes (MNB) [22].

Support Vector Machine (SVM) algorithms are a type of supervised learning algorithms that can be employed for both classification and regression purposes. SVMs have been applied successfully in a variety of classification problems such as text classification, image classification and recognizing hand-written characters. SVMs are famous for their classification accuracy and the ability to deal with high dimensional data [23]. One of the most important aspects of SVMs is selecting a suitable kernel function. A kernel function takes data as input and transforms it into the required form. There are different kernel functions like linear, polynomial, Radial Basis Function (RBF) and sigmoid. In our work, we selected RBF because of low time complexity. Also, RBF works quite well in practice, and it is relatively easy to tune as opposed to other kernels.

Decision Tree (DT) algorithm belongs to the family of supervised learning algorithms. Similar to SVM algorithms DT can be applied for both regression and classification problems. It is based on building a model that can predict class or value of the target variable by learning decision rules inferred from training data. The main advantages of DT cause to be selected in our work are (1) its ability to discover nonlinear relationships and interactions (2) interpretability (3) its robustness of dealing with outliers and missing data [24].

Random Forest (RF) algorithm is an ensemble learning method for classification and regression tasks. Generally, this classifier builds several decision trees on randomly selected sub-samples of training data. It then merges the results from different decision trees to make a decision about the final class of the test example. The process of voting helps to reduce the risk of overfitting. As a result, it improves classification accuracy. According to the above discussion we selected RF in our experiments [25].

AdaBoost (AB) classifier is another ensemble classifier that aims to build a robust classifier from a number of weak classifiers. Its process starts by building a model on the original dataset and then the subsequent classifiers attempts to correct the previous models [26].

Multilayer Perceptron (MLP) is a kind of neural network algorithms, and it is based on a network of perceptrons which organized in a feedforward topology. Basically, it consists of at least three layers: an input layer, a hidden layer, and an output layer. MLP belongs to the family of nonlinear classifier because it uses nonlinear activation functions in all layers except for the input layer [27]. We selected MLP because its nonlinear nature makes it suitable to learn and model complex relationships which are too complicated to be noticed by human or other learning algorithms.

mmeybodi@aut.ac.ir

5 Feature Engineering

The success of supervised machine learning algorithms strongly depends on how data is represented to them in terms of features. Feature engineering is the process of transforming raw data into features that make machine learning algorithms work better. In fact, providing an appropriate set of features is a fundamental issue of every learning algorithm. In this section, we will explore our features in general and our strategy for selecting suitable features [28].

Overall, we considered eight factors to extract feature out of them: (1) rate (2) number of votes (3) size (4) number of downloads (5) detailed permissions (6) general permissions (7) description (8) category. To identify the best set of features that results in the maximum classification accuracy and analyzing the role of each feature in predicting others, we accounted for every possible combination of features in our experiments. In order to calculate combinations for predicting a target variable *t*, all the features except *t* have been passed to a function which outputs every possible subset of features that could be constructed from the input features. That means for predicting target variable *t* if we pass N features to the function it will return all possible 2^N subsets of features. Furthermore, to use generated subsets in our experiments they must be converted into the vector space model. In the following, we will explain features in detail in terms of definition, idea and the process of converting to vectors.

Rate (R) users who install an application can score it from 1 to 5. The average rate is calculated by summing all the scores and then dividing it by the total number of participants. In order to avoid exceeding the number of possible values for this factor, we discriminated it by the procedure in Sect. 3.

Number of voters (RN) to distinguish between applications with a high number of voters and those with a low number of voters, we defined the multiplication of rate and number of voters as a single feature. This will help to reduce the effects of the rate for an application when few users rated it.

Size (S) we selected this factor as a feature because it might be related to the application category. For example, an application with high size might be a game rather than others. In order to avoid enormous number of possible values for this factor, we classified it by the process explained in Sect. 3.

Number of downloads (I) downloads count is another factor that could be correlated to others features. For example apps in popular categories could have a higher chance to be downloaded. So we used it as a feature in our final feature vector.

Description (D) Google Play allows developers to write about their apps. Regularly, this description talks about the features that users will get from the app. In the context of Natural Language Processing (NLP), the description might contain words which could represent the underlying problem better for our learning algorithms. In this matter, we selected description as a factor to extract feature out of it. To use the description as a feature, we converted preprocessed texts into vectors using x-square algorithm and TF-IDF as a weighting schema. Among those features, we selected the first top 300 features to build the vectors. The more features are included in the training phase, the more time consumes for training though. We also repeated our experiments

mmeybodi@aut.ac.ir

with the first top 100, 200 and 300 features in order to analyze the effects of description vector size on the final results of classifiers.

Permissions (P) every application gets various permissions on the host device. This feature might help effectively in the prediction of category or number of downloads. Thus, we decided to include this feature to the developed vector. Besides, every application has two kinds of permissions: General (GP) and Detailed (DP). For instance, an application could get four GPs: Location, Photos/Media/Files, Storage and Other. Every GP might have one or more DPs. For example, "Storage" might contain two detailed ones (1) Read the content of your USB storage, and (2) Modify or delete the contents of USB storage. We included both kinds of permissions in the vector. All crawled applications have 16 unique GPs and 199 unique DPs.

Category (C) each application's category is included in the vector. As it is shown in Table 2, we end up with 11 categories for all applications. So this number is involved in our vector as the last feature.

1	1	1	1	*	16	199	1
R	RN	S	I	D	GP	DP	C

Fig. 1. The features of the final vector and their size. **D stands for Description and the size varies with 100, 200 and 300.*

The final vector is shown in Fig. 1. The size of all features except D is 220. Thus, the total size will vary in the range of 320, 420 and 520.

6 Evaluation

6.1 Experimental Setup

To perform our experiments, we used Python which is a widely used open source programming language. As python has many different libraries to deal with machine learning and NLP problems, it could be used effectively for such processes. Nltk[2] and Sklearn[3] are two libraries that have been used in our work.

6.2 Results and Discussions

In this section, we report the experiment performance with respect to the feature sets. Our objective is to detect the correlation between features as well as detection of the best algorithm for predicting. For this purpose, we first tried to predict category of each application with the description of that app. Table 3 shows the result of this experiment. Note that P stands for precision, R for Recall and F for F-measure.

[2] Nltk.org.

[3] Scikit-learn.org.

mmeybodi@aut.ac.ir

Table 3. Prediction of category

	Algorithm	Features	D vector length	P	R	F
Predicting category	MLP	D	100	0.49	0.50	0.50
	SVM	D	100	0.51	0.46	0.48
	BNB	D	100	0.46	0.46	0.46
	MNB	D	100	0.46	0.46	0.46
	RF	D	100	0.44	0.45	0.44
	AB	D	100	0.43	0.43	0.43
	DT	D	100	0.39	0.39	0.39
	GNB	D	100	0.48	0.22	0.30

Based on the results, we can observe that GNB is not suitable for predicting this feature while MLP performs the best. In the next step, one variable has been changed which is the length of the description vector. This attribute changed to 200 and 300 in order to identify the influence of this element on the final results. The results are represented in Table 4 for the best classifiers from Table 3. The results show that increasing the size of the description vector leads to significant improvement in F-measure.

Table 4. Prediction of category with various sizes of description vector

	Algorithm	Features	D vector length	P	R	F
Predicting category	MLP	D	100	0.49	0.50	0.49
		D	200	0.59	0.59	0.59
		D	300	0.62	0.62	0.62
	SVM	D	100	0.51	0.46	0.48
		D	200	0.59	0.55	0.57
		D	300	0.60	0.56	0.58
	BNB	D	100	0.46	0.46	0.46
		D	200	0.55	0.54	0.54
		D	300	0.58	0.56	0.57
	MNB	D	100	0.46	0.46	0.46
		D	200	0.57	0.55	0.56
		D	300	0.60	0.58	0.59

The next step of our experiment is to predict category with various features. These features have been explained in Sect. 5. The results shown in Table 5 illustrate that the MLP algorithm performs better than any other algorithms in category predicting. Additionally, a simple comparison between two latter tables demonstrates that involving other features in the learning process leads to even better results. For example, using detailed permissions along with description improves the results in terms of f-measure by approximately 3%. This is probably because of some categories

mmeybodi@aut.ac.ir

that need special permissions, and this will help the classifier to distinguish between them. The results in Table 5 are selected among more than 700 experiments including all algorithms and features and indicate that other algorithms could not outperform MLP even with more features.

Table 5. Top ten results of predicting category with various algorithms and features

	Algorithm	Features	D vector length	P	R	F
Predicting category	MLP	D, P	300	0.64	0.64	0.64
	MLP	D, S, I, P	300	0.64	0.64	0.64
	MLP	D, R, S, P	300	0.64	0.64	0.64
	MLP	D, R, I, P	300	0.64	0.63	0.63
	MLP	D, R, P	300	0.64	0.63	0.63
	MLP	D, R, S, I, P	300	0.64	0.63	0.63
	MLP	D, I, P	300	0.63	0.63	0.63
	MLP	D, S, P	300	0.63	0.63	0.63
	MLP	D, R	300	0.62	0.62	0.62
	MLP	D, S	300	0.62	0.62	0.62

More precisely, after having done all these experiments for predicting categories, we expanded our study by predicting other features, namely Rate, Size and Install count. Table 6 reveals these results. Overall, RF and DT performed better than other algorithms in terms of predicting Size, Rate and Install count. For predicting the size of apps, the table shows that presence or absence of description in feature vector cannot affect the f-measure. This conclusion is based on the fifth row of the table which is similar to the other rows in terms of f-measure but has no description. Moreover, increasing the description vector length does not affect results. Comparing these results to other results shows that feature P can solely affect size prediction by 35%.

Additionally, in Rate and Install count prediction some experiments have been found without description which demonstrate the influence of other features. What's more, RN plays a significant role in predicting rate which is probably because of a strong correlation between these two factors. Besides, an interesting fact is that the reduction of feature vector size does not necessarily lead to f-measure plummet. For example in install count prediction, a tiny vector (R, RN, S, and C) with the size of 4 with RF, performed as good as huge vectors with DT algorithm.

We extended our experiments even more in order to analyze to what extent different general permissions have correlation with each other. As it is mentioned in Sect. 5, we found 16 different general permissions in our dataset. To perform correlation analysis, we first analyze the distribution of each general permission. Then we removed permissions which had an extremely unbalanced proportion. To achieve this, we ignored the permissions with less than 1000 sample of each label. Finally, we ended up with eight general permissions. Table 7 shows the distribution of these eight permissions which are considered to be involved in our experiments.

mmeybodi@aut.ac.ir

Table 6. Top 10 results of predicting size, rate and install count with various algorithms and features

	Algorithm	Features	D vector length	P	R	F
Predicting size	RF	D, R, I, P, C	200	0.39	0.44	0.41
	RF	D, R RN, P, C	300	0.39	0.44	0.41
	RF	D, I, P, C	100	0.39	0.43	0.40
	RF	D, R, P, C	100	0.38	0.44	0.40
	RF	R, RN, I, P, C	–	0.38	0.43	0.40
	RF	D, P	100	0.38	0.43	0.40
	RF	D, I, P	100	0.38	0.43	0.40
	RF	D, P, C	100	0.38	0.43	0.40
	RF	D, R, RN, P	100	0.38	0.43	0.40
	RF	D, RN, P, C, S	100	0.38	0.43	0.40
Predicting rate	RF	RN, S, I, P, C	–	0.56	0.58	0.57
	RF	D, RN, S, I	200	0.55	0.58	0.56
	RF	D, RN, I, P	100	0.55	0.57	0.56
	RF	D, RN, S, I, P	100	0.55	0.57	0.56
	RF	D, RN, I, P, C	100	0.55	0.57	0.56
	RF	RN, S, I, P	–	0.55	0.56	0.55
	RF	D, RN, I, C	100	0.54	0.56	0.55
	RF	D, RN, S, I, C	100	0.54	0.56	0.55
	DT	D, RN, I, P	200	0.54	0.54	0.54
	DT	D, RN, S, I, P, C	100	0.54	0.54	0.54
Predicting install count	RF	R, RN, S, C	–	0.65	0.64	0.64
	DT	D, R, RN, P, C	100	0.64	0.64	0.64
	RF	D, R, RN, C	100	0.64	0.63	0.63
	DT	D, R, RN, P	200	0.64	0.63	0.63
	DT	D, R, RN, P, C	300	0.64	0.63	0.63
	DT	D, R, RN, P, C	200	0.64	0.63	0.63
	DT	D, R, RN, S, P, C	200	0.63	0.63	0.63
	DT	D, R, RN, P	300	0.63	0.63	0.63
	DT	D, R, RN, S, P, C	100	0.63	0.63	0.63
	DT	D, R, RN, S, P	200	0.63	0.63	0.63

In order to determine the correlations between selected general permissions, we apply the same approach as we discussed earlier in this section. However, in the feature selection part we considered all the general permissions except the target one in the feature vector; e.g., if GP1 is the target for prediction then other seven general permissions (i.e. GP2, GP3, GP4, GP5, GP6, GP7, and GP8) will make the feature vector. For classification, we used the same algorithms as before. Table 8 shows our results in predicting different general permissions with eight classification algorithms in terms of precision, recall, and F-measure.

mmeybodi@aut.ac.ir

Table 8 reveals several interesting points. First of all, despite the unbalanced distribution of classes, the classification results are substantially promising. In contrast to the Table 5, where there was a significant difference between the results of one special classification algorithm compare to the other algorithms, the results of Table 8 demonstrate that several classification algorithms could achieve higher results in comparison with the rest in terms of P, R, and F.

For predicting GP1, all the classification algorithms have the same performance (99%) based on all evaluation metrics. Performance of the classification algorithms for GP2 is relatively similar except for AdaBoost Algorithm, where there is a minor

Table 7. Different permissions and their distribution

Id	Target permission	1	0
GP1	Device & App History	3426 (53%)	3242 (47%)
GP2	Contact	4696 (70%)	1972 (30%)
GP3	Photos | Media | Files	1024 (15%)	5163 (85%)
GP4	Calendar	1631 (24%)	5037 (76%)
GP5	Cellular data settings	1857 (28%)	4811 (72%)
GP6	Wearable sensors | Activity data	1529 (23%)	5139 (77%)
GP7	Storage	1918 (29%)	4750 (71%)
GP8	Microphone	4695 (70%)	1973 (30%)

increase in precision (1%) as opposed to the other algorithms. The best results in GP3 classification are same as the best results in GP2. However, there are more than one algorithms which have reached the best performance. Results in predicting GP4 show that four algorithms have performed better in comparison to others.

F-measure scores in predicting GP5 are quite similar (99%) in most algorithms. However, there are three algorithms that have done slightly better with 1% percent increase in P and R. Performance of the algorithms in classifying apps when GP6 is the target are the highest among all the classifications results in Table 8 with 100% for all evaluation metrics. That means, there is a strong correlation between GP6 and other selected general permissions. Best results in predicting GP7 are completely equal with the best in GP1 and GP5. Comparing to other results in Table 8, the performance of predicting GP8 has significantly decreased by almost 15% but still promising.

Finally, based on the classifications results and the above discussions we believe there are strong correlations between all the selected general permissions which leads to high-performance results in predicting each general permissions with the rest.

6.3 Practical Usage

The practical usage of our analysis would be in several domains. In case of predicting category, there are apps which have been miscategorized by developers in the Google play store [5]. Therefore, concerning the supervised learning algorithms as solutions for the problem, the type of features which are selected in training phase would be crucial.

mmeybodi@aut.ac.ir

Table 8. Prediction results for eight different GPs.

		P	R	F		P	R	F		P	R	F		P	R	F
BNB	GP1	**0.99**	**0.99**	**0.99**	GP2	0.97	0.97	0.97	GP3	**0.98**	**0.97**	**0.97**	GP4	0.95	0.91	0.93
GNB		**0.99**	**0.99**	**0.99**		0.97	0.97	0.97		0.97	0.97	0.97		0.95	0.94	0.95
MNB		**0.99**	**0.99**	**0.99**		0.97	0.97	0.97		0.87	0.87	0.85		0.94	0.97	0.96
DT		**0.99**	**0.99**	**0.99**		0.97	0.97	0.97		**0.98**	**0.97**	**0.97**		**0.95**	**0.97**	**0.96**
RF		**0.99**	**0.99**	**0.99**		0.97	0.97	0.97		**0.98**	**0.97**	**0.97**		**0.95**	**0.97**	**0.96**
MLP		**0.99**	**0.99**	**0.99**		0.97	0.97	0.97		**0.98**	**0.97**	**0.97**		**0.95**	**0.97**	**0.96**
AB		**0.99**	**0.99**	**0.99**		**0.98**	**0.97**	**0.97**		**0.98**	**0.97**	**0.97**		**0.95**	**0.97**	**0.96**
SVM		**0.99**	**0.99**	**0.99**		0.97	0.97	0.97		**0.98**	**0.97**	**0.97**		0.94	0.97	0.96
BNB	GP5	**0.99**	**0.99**	**0.99**	GP6	1.0	0.99	0.99	GP7	**0.99**	**0.99**	**0.99**	GP8	0.84	0.82	0.83
GNB		0.99	0.93	0.95		1.0	0.86	0.92		**0.99**	**0.99**	**0.99**		0.83	0.87	0.84
MNB		0.98	0.99	0.99		**1.0**	**1.0**	**1.0**		0.96	0.96	0.96		0.79	0.89	0.84
DT		**0.99**	**0.99**	**0.99**		**1.0**	**1.0**	**1.0**		**0.99**	**0.99**	**0.99**		0.83	0.89	0.84
RF		**0.99**	**0.99**	**0.99**		**1.0**	**1.0**	**1.0**		**0.99**	**0.99**	**0.99**		**0.86**	**0.89**	**0.84**
MLP		0.98	0.99	0.99		**1.0**	**1.0**	**1.0**		**0.99**	**0.99**	**0.99**		0.84	0.89	0.84
AB		0.98	0.99	0.99		**1.0**	**1.0**	**1.0**		**0.99**	**0.99**	**0.99**		0.83	0.89	0.84
SVM		0.98	0.99	0.99		**1.0**	**1.0**	**1.0**		**0.99**	**0.99**	**0.99**		0.79	0.89	0.84

For instance, in our analysis we figured out that description and permission are more important for predicting app's category. Additionally, another solution is to use clustering techniques [16] or the concepts of graph theory (community detection). In both cases, description and permission could be applied as a feature vector since they are highly correlated with app's category based on our analysis.

Predicting size is a relatively small and new research area [3]. However, our analysis has shown us there are not strong correlations among other features and the app's size. Therefore, we do not claim that this analysis can certainly be helpful in this task.

In contrast, as far as the practical usage of finding correlations among general permissions are concerned, there are several methods that can be used to identify hazardous applications. To this aim, a clustering algorithm, like k-means, can find the applications with *suspicious* permissions; which are the applications that get a special permission which is not common among their similar apps. Furthermore, based on our results in predicting each general permission, we expect to obtain acceptable accuracy in identifying dangerous applications with unusual behavior.

7 Conclusion

This paper presents an immense study on classifying Google Play applications with various algorithms and multiple features. The first part of our experimental results on more than 7000 applications demonstrates that there is a significant difference between algorithms for predicting each feature. More precisely, the MLP algorithm outperforms the rest in term of category prediction, while Decision Tree and Random Forest are the best algorithms to predict other features: i.e. rate, size, and the number of installations. Also, our results show that increasing the feature vector size does not necessarily leads to better accuracy and it is possible to achieve the same f-measure with small vectors.

mmeybodi@aut.ac.ir

The second part of our experiments reveals that there are strong correlations among general permissions. Moreover, the performance of different algorithms in the second part were fairly same and considerably high. Future works would include correlation analysis between other general permissions that have not been covered here on a larger and more balanced dataset. Finally, the findings of the second part of our experiments could be used to propose more sophisticated methods in predicting apps that get suspicious permissions.

References

1. Google announces over 2 billion monthly active devices on Android. https://www.theverge.com/2017/5/17/15654454/android-reaches-2-billion-monthly-active-users. Accessed 12 Aug 2018
2. Google Play Store: number of apps 2018—Statistic. https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/. Accessed 12 Aug 2018
3. Martin, W., Sarro, F., Jia, Y., Zhang, Y., Harman, M.: A survey of app store analysis for software engineering. IEEE Trans. Softw. Eng. **43**(9), 817–847 (2017)
4. Radosavljevic, V., et al.: Smartphone app categorization for interest targeting in advertising marketplace. In: Proceedings of the 25th International Conference Companion on World Wide Web - WWW 2016 Companion, pp. 93–94 (2016)
5. Surian, D., Seneviratne, S., Seneviratne, A., Chawla, S.: App miscategorization detection: a case study on Google Play. IEEE Trans. Knowl. Data Eng. **29**(8), 1591–1604 (2017)
6. Berardi, G., Esuli, A., Fagni, T., Sebastiani, F.: Multi-store metadata-based supervised mobile app classification. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC 2015, pp. 585–588 (2015)
7. Cunha, A., Cunha, E., Peres, E., Trigueiros, P.: Helping older people: is there an app for that? Procedia Comput. Sci. **100**, 118–127 (2016)
8. Olabenjo, B.: Applying Naive Bayes Classification to Google Play Apps Categorization, August 2016
9. Liu, M., Wang, H., Guo, Y., Hong, J.: Identifying and analyzing the privacy of apps for kids. In: Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications - HotMobile 2016, pp. 105–110 (2016)
10. Wang, H., Li, Y., Guo, Y., Agarwal, Y., Hong, J.I.: Understanding the purpose of permission use in mobile apps. ACM Trans. Inf. Syst. **35**(4), 1–40 (2017)
11. Wu, D.-J., Mao, C.-H., Wei, T.-E., Lee, H.-M., Wu, K.-P.: DroidMat: android malware detection through manifest and API calls tracing. In: 2012 Seventh Asia Joint Conference on Information Security, pp. 62–69 (2012)
12. Varma, P.R.K., Raj, K.P., Raju, K.V.S.: Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms. In: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 294–299 (2017)
13. Gorla, A., Tavecchia, I., Gross, F., Zeller, A.: Checking app behavior against app descriptions. In: Proceedings of the 36th International Conference on Software Engineering - ICSE 2014, pp. 1025–1035 (2014)
14. Ma, S., Wang, S., Lo, D., Deng, R.H., Sun, C.: Active semi-supervised approach for checking app behavior against its description. In: 2015 IEEE 39th Annual Computer Software and Applications Conference, pp. 179–184 (2015)

mmeybodi@aut.ac.ir

15. Shabtai, A., Fledel, Y., Elovici, Y.: Automated static code analysis for classifying android applications using machine learning. In: 2010 International Conference on Computational Intelligence and Security, pp. 329–333 (2010)
16. Al-Subaihin, A.A., et al.: Clustering mobile apps based on mined textual features. In: Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM 2016, pp. 1–10 (2016)
17. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
18. Kotsiantis, S.: Supervised machine learning: a review of classification techniques. In: Emerging Artificial Intelligence Applications in Computer Engineering, pp. 3–24 (2007)
19. Kotsiantis, S., Panayiotis, P.: Recent advances in clustering: a brief survey. WSEAS Trans. Inf. Sci. Appl. **1**(1), 73–81 (2004)
20. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
21. Lewis, D.D.: Naive (Bayes) at forty: the independence assumption in information retrieval, pp. 4–15. Springer, Heidelberg (1998)
22. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: AAAI 1998 Workshop on Learning for Text Categorization, vol. 752, no. 1, pp. 41–48 (1998)
23. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Proceedings of the 10th European Conference on Machine Learning, pp. 137–142. Springer (1998)
24. Rokach, L., Maimon, O.: Top-down induction of decision trees classifiers—a survey. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **35**(4), 476–487 (2005)
25. Svetnik, V., Liaw, A., Tong, C., Culberson, J.C., Sheridan, R.P., Feuston, B.P.: Random forest: a classification and regression tool for compound classification and QSAR modeling. J. Chem. Inf. Comput. Sci. **43**(6), 1947–1958 (2003)
26. Freund, Y., Schapire, R., Abe, N.: A short introduction to boosting. J.-Jpn. Soc. Artif. Intell. **14**(771–780), 1612 (1999)
27. Gardner, M.W., Dorling, S.R.: Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmos. Environ. **32**(14–15), 2627–2636 (1998)
28. Dong, G., Liu, H.: Feature Engineering for Machine Learning and Data Analytics. CRC Press, Boca Raton (2018)

mmeybodi@aut.ac.ir



Information Verification Enhancement Using Entailment Methods

Arefeh Yavary[1], Hedieh Sajedi[1(✉)], and Mohammad Saniee Abadeh[2,3]

[1] Department of Computer Science, School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, Iran
{yavary_rf, hhsajedi}@ut.ac.ir

[2] Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran
saniee@modares.ac.ir

[3] School of Computer Science, Institute for Research in Fundamental Science (IPM), Tehran, Iran

Abstract. Information verification is a hot topic, especially because of the fact that the rate of information generation is so high and increases every day, mainly in social networks like Twitter. This also causes social networks be invoked as a news agency for most of the people. Accordingly, information verification in social networks becomes more significant. Therefore, in this paper a method for information verification on Twitter is proposed. The proposed method for Tweet verification is going to employ textual entailment methods for enhancement of previous verification methods on Twitter. Aggregating the results of entailment methods in addition to the state-of-the-art methods, can enhance the outcomes of tweet verification. Also, as writing style of tweets is not perfect and formal enough for textual entailment, we used the language model to supplement tweets with more formal and proper texts for textual entailment. Although, singly utilizing of entailment methods for information verification may result in acceptable results, it is not possible to provide relevant and valid sources for all of the tweets, especially in early times by posting tweets. Therefore, we utilized other sources like as a User Conversational Tree (UCT) besides utilizing entailment methods for tweet information verification. The analysis of UCT is based on the pattern extraction from the UCT. Experimental results indicate that using entailment methods enhances tweet verification.

Keywords: Information verification · Textual entailment

1 Introduction

These days, as we refer to social networks, we face to several messages which we not sure, do we trust or believe in them or not. This distrust makes the social networks as the unpleasant environment, especially in a crisis, which result in concern between people. Also, by the high rate of data generation in social networks like as Twitter, this social media is generally used for getting news. Hence, it's vital and important to check

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 217–225, 2020.
https://doi.org/10.1007/978-3-030-37309-2_17
mmeybodi@aut.ac.ir

the validity of information which spread over the Twitter. Despite the reasons mentioned, in this paper, we are going to check the validity of tweets. By now, some approaches suggested for rumor detection of Tweets, also rumor diffusion is studied in some cases, too. The main challenge of rumor detection is that there is not available some reliable and credible source for determining the validation of a tweet, in all cases. Therefore, in our proposed method, we consider two different sources for checking the validity of a tweet. In our proposed method of this paper, we are going to get better income in information validity of tweets. Therefore, we aggregate the textual entailment methods with the results of analysis in a UCT of intended tweet for checking validity in order to enhance the results of information validation. This aggregating is done using a weighted voting classifier on the result of entailment on the tweet and some references and the analysis of the belonging UCT. Furthermore, in our suggested method, we faced with several challenges. The most important one is that, as the context and writing style of tweets are tidy and also the length of tweets are short, it is hard to get worthy outcomes in using tweet in textual entailment methods. Hence, we used a language model in order to make tweets language style, more acceptable. In overall, our contribution in this paper for information validation in twitter are:

- Using textual entailment to enhance rumor detection on Twitter
- Using a language model for making tweets more acceptable in writing style
- Consider subtree in analyzing UCT
- Propose a weighted voting classifier in order to aggregate the result of entailment method and UCT analysis

In our experiments, we used the just available public benchmark data set for rumor detection on Twitter. The experimental result shows that our proposed method improved the result of information validation in Twitter with respect to other proposed method which tested on the benchmark. Also, the results show that entailment methods boost the results of information validation. Also, results of information validation using textual entailment are very astonishing. But, as maybe it is not possible to collect valid information sources for all of the tweets, textual entailments must be used in combination with other methods of information verifications like as what we used a weighted voting classifier to aggregate the result of UCT analysis and textual entailment.

In the subsequent of paper, first we review the related works of rumor detection, textual entailment and voting classifiers. Then, some preliminary knowledge has been stated before expressing suggested approach. After that the results and discussion come to account. At the end, we conclude in conclusion.

2 Literature Review

In this part, recent studies in information verification, textual entailments and voting classifier are reviewed.

mmeybodi@aut.ac.ir

2.1 Information Verification

Ma et al. [8], used multi-task learning based on the neural framework for rumor detection. Thakur et al. [9], studied rumor detection on Twitter using supervised learning method. Rumor diffusion investigated by Li et al. [10]. Majumdar et al. [11], proposed a method for rumor detection in financial scope using a high volume of data. Mondal et al. [12], focused on the fast and timely detection of rumor, which is so important in disaster and crisis.

2.2 Textual Entailment

In textual entailment methods, we have a hypothesis H and theory T. Our task is to decide whether is we can entail H from the T or not. This task can have three or two labels: entailment/non-entailment/contradiction or positive/negative. Where positive class equals to entailment and negative class equals with non-entailment and contradiction. Entailment means when we can entail T from H, contradiction means H contradicts T and non-entailment means when H get any conclusion about T [3]. Silva et al. [3], proposed a textual entailment method using the definition graph. Rocha et al. [4] and Almarwani et al. [6], studied textual entailments in Portuguese and Arabic languages, respectively. Balazs et al. [5], suggested a representation method for sentences to be used for textual entailment in attention based neural networks. Burchardt et al. [7], annotated a textual entailment corpus using FrameNet.

2.3 Ensemble Classifier

Ensemble classifiers are a family of classifiers which ensemble a number of classifiers with an aggregating method to aggregate the results of the classifier to get a better classifier. The difference between different ensemble classifiers are the difference between their method of ensemble [2]. Onan et al. [18] used weighted voting classifier with combination of differential evolution method for sentiment analysis. An online-weighted ensemble classifier for the geometrical data stream is suggested by Bonab et al. [2], Gu et al. [13], proposed a rule-based ensemble classifier for remote sensing scenes.

2.4 Extreme Learning Machine

Extreme Learning Machine (ELM) is a single hidden layer neural network, so it has three layers: input layer, hidden layer and output layer. The special property of ELM is its interesting learning method as follows: first, the weights between input and hidden layer is set randomly. Then, using some kind of matrix transpose, the weights of edges between hidden and output layer are computed. ELM can use different activation function and has several extensions like as a kernel extension [16] or multilayer ELM, which has more than one hidden layer [17].

mmeybodi@aut.ac.ir

3 Proposed Approach

As the different challenges in rumor detection, we used result in rumor detection for analysis of two sources: 1- textual entailment method on source news, 2- analysis of UCT. Then for each of these two sources, we train two classifiers separately, and after that, by using weighted ensemble voting classifier, we ensemble the results of these two separated classifiers to create a new classifier. The process of our approach is illustrated in Fig. 1. Each of the sub-process of the method is expressed in the following.

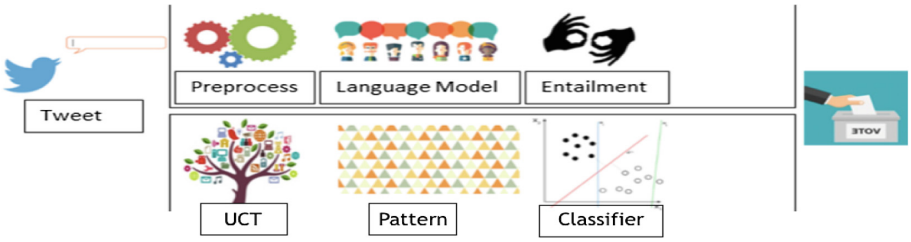


Fig. 1. The architecture of proposed approach.

3.1 Entailment-Based Classifier

In this part, modeling the entailment classifier is described. First, the tweets are corrected by language modeling, then the textual entailment is utilized on the tweets: *1- Formal modeling of tweet*: As the tweets are short and concise, they are not following the formal English writing style. Because of that, we used a language model [14], to correct the tweet writing style. *2- Textual Entailment*: The used entailment methods are as follows [15]: Ed-RW (Edit distance comp: Fixed weight lemma RES: wordnet), M-TVT (MaxEntClassification COMP: TreeSkeleton RES: VerbOcean, TreePattern), M-TWT (MaxEntClassification COMP: TreeSkeleton RES: WordNet, TreePattern), M-TWVT (MaxEntClassification COMP: TreeSkeleton RES: WordNet, VerbOcean, TreePattern), PRPT (P1EDA RES: Paraphrase table).

3.2 UCT-Based Classifier

UCT has several usage in different cases. According that we working on the context of Twitter, we define UCT in use of Twitter. Our intended UCT [1], is structured as follows: First, the tweet which we want to analyze it for veracity is put to the root of the tree. Then, any reply to this tweet is considered as the children of the root. Then, any undirected reply is considered as the child of the tweet which we reply to. In this way, the UCT is created. After that, each reply in the UCT is labeled corresponding to the its opinion with respect to the main tweet and its parents. These labels are: Support, Deny, Query and Comment. Support and Deny means that the reply agrees and disagrees with the corresponding tweet, respectively. Query means that the reply asks for some reference about the main tweet. A Comment means that the reply tweet just gives some comment, without any indication to deny or support of the tweet. For analyzing UCT,

mmeybodi@aut.ac.ir

two groups of patterns of branched and un-branched patterns are proposed as the following: 1- Un-branched subtree: As the name of the pattern shows, these group of patterns are subtrees of patterns which has not any branches. Like as N-gram. 2- Branched subtree: These patterns are which already has at least one branch.

3.3 Weighted Voting Ensemble Classifier

In this phase, the result of Entailment-based classifier and UCT-based classifier is aggregated using a weighted voting classifier. We used Grid Search for tuning weights of voting classifier [19].

4 Experiments and Discussion

In this section, first the experimental environment is explained. Then the proposed method is compared with other methods. Then, the experimental results are discussed.

4.1 Experimental Environment

The dataset is defined in [1], also this data set is just publicly available dataset for rumor detection. The train set contains 137, 62 and 98 tweets for True, False and Unverified labels, respectively. The test set contains 8, 12 and 8 tweets for True, False and Unverified labels, respectively. Preprocessing, UCT analysis and ensemble voting classifier are implemented in Python. Textual entailment method is implemented in Java. ELM method is implemented in MATLAB 2016.a.

4.2 Comparing Proposed Methods with Other Methods

Methods of comparison are the systems introduced in [1]. Evaluation measures are same as what used in [1]. These measures are Score, Confidence RMSE and Final Score. The Score is same as accuracy, Confidence RMSE is measured to compute value of confidence error and Final Score is computed by multiplying of Score and (1-Confidence RMSE).

4.3 Results and Discussion

In the following tables, results of just using entailment methods, best results of using UCT patterns in train and test set, results of combination of entailment methods and patterns are represented in Tables 1, 2, 3 and 4, respectively. In each of the mentioned tables, best results are shown in bold case. In Tables 3 and 4, the results of systems for comparison are shown in gray cell. As Table 1 shows, best result for textual entailment is P1EDA RES: Paraphrase table method. Although a recent Twitter increase the length of tweets because of the fact that some languages has larger encoding of characters, but this results in longer tweets which make an easier analysis of tweets in textual entailment. Also, in analyzing UCT, as tweets are short, we can suppose that each tweet just gets one of the labels of Support, Deny, Query and Comment. Also, as the

mmeybodi@aut.ac.ir

normalizing the patterns with the maximum length of the pattern category could be useful for affect the long patterns, too. Parameters like as the time interval between posting replies could be an important feature, too. In Fig. 2, the diagram for comparison of different entailment methods is illustrated. In Fig. 3, different rumor detection methods are comprised.

Table 1. Results of using entailment methods.

Approach	Evaluation measures		
	Score	Confidence RMSE	Final score
Ed-RW	*0.778*	*0.947*	*0.041*
M-TVT	*0.445*	*0.929*	*0.032*
M-TWT	*0.445*	*0.925*	*0.033*
M-TWVT	*0.445*	*0.934*	*0.030*
PRPT	***0.778***	***0.629***	***0.289***

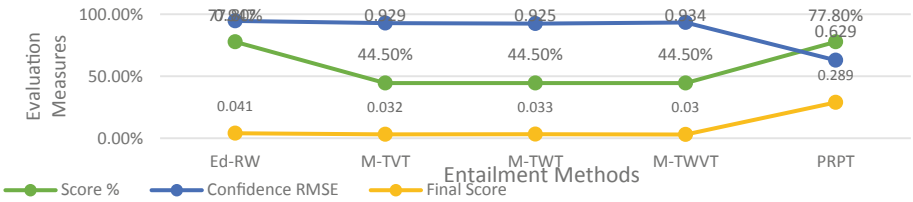


Fig. 2. The diagram for comparison of different entailment methods.

Table 2. Best results of using patterns of UCT in train set.

Approach	Evaluation measures		
	Score	Confidence RMSE	Score %
Elm-kernel (RBF Kernel)	*0.960*	*0.063*	*0.900*
Elm-kernel (Linear Kernel)	*0.467*	*0.846*	*0.072*
Elm-sine	*0.971*	*0.037*	*0.935*
Elm-rbfs	*0.960*	*0.063*	*0.900*
Elm-sine	*0.971*	*0.037*	*0.935*
Elm-rbfs	*0.960*	*0.063*	*0.900*
Elm-tribas	*0.971*	*0.037*	*0.935*
Multinominal Naive Bayes	*0.684*	*0.478*	*0.357*
Support Vector Machine	*0.820*	*0.180*	*0.672*
Multi-Layer Perceptron	*0.184*	*0.816*	*0.034*

mmeybodi@aut.ac.ir

Table 3. Best results of using patterns of UCT in test set.

Approach	Evaluation Measures		
	Score	Confidence RMSE	Final Score
Elm-kernel (RBF Kernel)	*0.536*	*0.607*	*0.210*
Elm-kernel (Linear Kernel)	*0.321*	*0.679*	*0.103*
Elm-sig	***0.642***	***0.301***	***0.424***
Elm-hardlim	*0.607*	*0.536*	*0.282*
Elm-sine	***0.642***	***0.301***	***0.424***
Elm-rbfs	*0.607*	*0.536*	*0.282*
Elm-tribas	*0.643*	*0.301*	*0.424*
Multinominal Naive Bayes	*0.500*	*0.679*	*0.161*
Support Vector Machine	*0.429*	*0.571*	*0.184*
Multi-Layer Perceptron	*0.500*	*0.679*	*0.161*
DFKI DKT	*0.393*	*0.845*	*0.061*
ECNU	*0.464*	*0.736*	*0.122*
IITP	*0.286*	*0.807*	*0.055*
IKM	*0.536*	*0.736*	*0.142*
NileTMRG	*0.536*	*0.672*	*0.176*
Baseline	*0.571*	-	-

Table 4. Results in combination using of entailment and patterns.

Approach	Evaluation Measures		
	Score	Confidence RMSE	Final Score
Elm+Entailment	***0.714***	***0.401***	***0.428***
DFKI DKT	0.393	0.845	0.061
ECNU	0.464	0.736	0.122
IITP	0.286	0.807	0.055
IKM	*0.536*	*0.736*	*0.142*
NileTMRG	*0.536*	*0.672*	*0.176*
Baseline	*0.571*	-	-

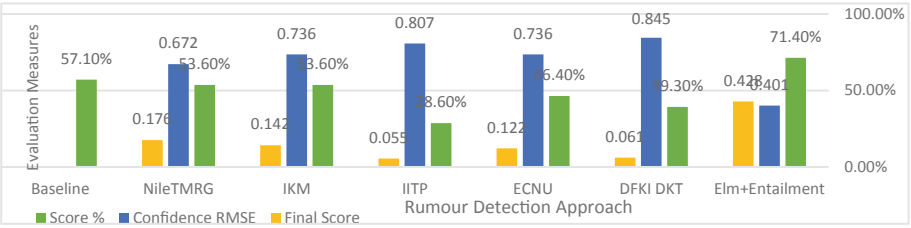


Fig. 3. The comparison of different rumor detection methods in different evaluation measures.

mmeybodi@aut.ac.ir

5 Conclusion and Future Works

Rumor detection is a hot and open research area. This research topic is very challenging, especially because there is no reliable source for determining the validity of all of the tweets. Also, these days rumor are mainly spreading through social networks. Between different social networks, Twitter is more disposed for rumor spreading, because of the high rate of information generation rate and the length of the tweet. Therefore, we selected Twitter as the social media for rumor detection study. By the challenge of rumor detection, we consider two kinds of resources for rumor detection, which are user-feedbacks and news resources. Our method is analyzing UCT and entailment method to considering the sources for rumor detection, respectively. Also, as tweets are somehow untidy, we used the language model to clean the tweets in entailment methods. Then the results of them are aggregated using an ensemble classifier. Experimental results of our method on the benchmarks in rumor detection show that our method has over passed the state of the art methods. To continue our method in the future, we propose to extend our method by studying more special patterns in UCTs and special entailment methods.

Acknowledgment. This research was in part supported by a grant from IPM. (No. CS1397-4-98).

References

1. Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Hoi, G.W.S., Zubiaga, A.: SemEval-2017 task 8: RumourEval: determining rumor veracity and support for rumours. In: Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval-2017, April 2017
2. Bonab, H.R., Can, F.: GOOWE: geometrically optimum and online-weighted ensemble classifier for evolving data streams. ACM Trans. Knowl. Discov. Data **12**(2), 1–33 (2018)
3. Silva, V.S., Freitas, A., Handschuh, S.: Recognizing and justifying text entailment through distributional navigation on definition graphs. In: Thirty-Second AAAI Conference on Artificial Intelligence AAAI, November 2017
4. Rocha, G., Cardoso, H.L.: Recognizing textual entailment: challenges in the Portuguese language. Information **9**(4), 76 (2018)
5. Balazs, J., Marrese-Taylor, E., Loyola, P., Matsuo, Y.: Refining raw sentence representations for textual entailment recognition via attention. In: Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, September 2017
6. Almarwani, N., Diab, M.: Arabic textual entailment with word embeddings. In: Proceedings of the Third Arabic Natural Language Processing Workshop, April 2017
7. Burchardt, A., Pennacchiotti, M.: FATE: annotating a textual entailment corpus with FrameNet. In: Handbook of Linguistic Annotation, pp. 1101–1118, June 2017
8. Ma, J., Gao, W., Wong, K.-F.: Detect rumor and stance jointly by neural multi-task learning. In: Companion of the The Web Conference 2018 - WWW 2018, April 2018
9. Thakur, H.K., Gupta, A., Bhardwaj, A., Verma, D.: Rumor detection on Twitter using a supervised machine learning framework. Int. J. Inf. Retrieval Res. **8**(3), 1–13 (2018)
10. Li, D., Gao, J., Zhao, J., Zhao, Z., Orr, L., Havlin, S.: Repetitive users network emerges from multiple rumor cascades. arXiv preprint arXiv:1804.05711 (2018)

mmeybodi@aut.ac.ir

11. Majumdar, A., Bose, I.: Detection of financial rumors using big data analytics: the case of the Bombay Stock Exchange. J. Organ. Comput. Electron. Commerce **28**(2), 79–97 (2018)
12. Mondal, T., Pramanik, P., Bhattacharya, I., Boral, N., Ghosh, S.: Analysis and early detection of rumors in a post disaster scenario. Inf. Syst. Front. **20**, 961–979 (2018)
13. Gu, X., Angelov, P.P., Zhang, C., Atkinson, P.M.: A massively parallel deep rule-based ensemble classifier for remote sensing scenes. IEEE Geosci. Remote Sens. Lett. **15**(3), 345–349 (2018)
14. Ng, A.H., Gorman, K., Sproat, R: Minimally supervised written-to-spoken text normalization. In: 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), December 2017
15. Magnini, B., Zanoli, R., Dagan, I., Eichler, K., Neumann, G., Noh, T.-G., Padó, S., Stern, A., Levy, O.: The excitement open platform for textual inferences. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, June 2014
16. Huang, G.-B.: An insight into extreme learning machines: random neurons, random features and kernels. Cogn. Comput. **6**(3), 376–390 (2014)
17. Yang, Y., Wu, Q.M.J.: Multilayer extreme learning machine with subnetwork nodes for representation learning. IEEE Trans. Cybern. **46**(11), 2570–2583 (2016)
18. Onan, A., Korukoğlu, S., Bulut, H.: A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. Expert Syst. Appl. **62**, 1–16 (2016)
19. Lavalle, S.M., Branicky, M.S.: On the relationship between classical grid search and probabilistic roadmaps. In: Springer Tracts in Advanced Robotics Algorithmic Foundations of Robotics V, pp. 59–75, August 2004

mmeybodi@aut.ac.ir



A Clustering Based Approximate Algorithm for Mining Frequent Itemsets

Seyed Mohsen Fatemi, Seyed Mohsen Hosseini, Ali Kamandi(✉), and Mahmood Shabankhah

School of Engineering Science, College of Engineering, University of Tehran, Tehran, Iran
{mohsen.fatemi,mohsen.hosseini72,kamandi,shabankhah}@ut.ac.ir

Abstract. We present an approximate algorithm for finding frequent itemsets. The main idea can be described as turning the problem of mining frequent itemsets into a clustering problem. More precisely, we first represent each transaction by a vector using one-hot encoding scheme. Then, by means of mini batch k-means, we group all transactions into a number of clusters. The center of each cluster can be assumed as a potential candidate for a frequent itemset. To test the validity of this assumption, we compute the support of itemsets represented by cluster centers. All clusters that do not meet the minimum support condition will be removed from the set of clusters. As our experiments show, this approximate algorithm can capture more than 90% of all frequent itemsets at a much faster rate than the competing algorithms. Moreover, we show that the execution time of our algorithm is linear.

Keywords: Apriori · FP-Growth · Frequent pattern mining · Association rule mining · Partition-based method · Mini-batch K-means

1 Introduction

Market basket analysis in the form of association rule mining was first proposed by Agrawal [1]. He analyzed customers shopping basket in order to find associations between the different purchased items by customers. And it becomes to be one of the most essential needs in data mining tasks because they can be used to find sequential patterns, correlations, particle periodicity, and classification or in other types of business applications.

With emergence of online stores the need for finding frequent itemsets in large datasets upsurged. Big companies like Amazon or eBay need to find this itemsets in faster time. Nonetheless Time complexity of finding frequent itemsets is still a challenge in the field.

To solve the time complexity of finding frequent itemsets in recent years a lot of algorithms have been proposed [2,8,11,13,17]. In this paper we introduce an approximate algorithm to find frequent itemsets. This algorithm is a clustering

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 226–237, 2020.
https://doi.org/10.1007/978-3-030-37309-2_18

mmeybodi@aut.ac.ir

based approach which uses mini batch k-means [14]. Each constructed cluster is a candidate to be frequent itemsets. To be assured of that we should count the number of appearance for each cluster in the dataset.

The rest of the paper is organized as follows. In Sect. 2, a formal definition of the problem is given. The related work is presented in Sect. 3. Section 4 gives an illustration of the main algorithm. The results obtained from simulations on the runtime and accuracy of our algorithm and that of FP-growth algorithm is presented in Sect. 5. Finally, Sect. 6 summarizes the results and offers some future research topics.

2 Problem Definition

Definition 1. *Let $I = \{i_1, \cdots, i_n\}$ be a set of items (i.e., products in a store). A nonempty subset $IS = \{i_j \in I : j : 1, \cdots, m\}$ of I is called an Itemset.*

Definition 2. *A transaction T is a pair $\langle tid, I \rangle$, where tid is the transaction identifier (each Transaction has a unique transaction ID) and I is an Itemset. A collection $D = \{t_1, \cdots, t_m\}$ of transactions is called a database [6].*

Definition 3. *The support of an itemset X, denoted by $supp(X)$, is the proportion of transactions T in the dataset D which contain X [6]. More precisely,*

$$supp(X) = \frac{|\{T \in D : X \subseteq T\}|}{|D|} \tag{1}$$

Definition 4. *An itemset X is a called a frequent pattern if its support is no less than a predefined threshold called minimum support. In other words, $X \subset I$ is a frequent pattern if $supp(X) \geq min_sup$.*

Definition 5. *An itemset X is a **maximal frequent itemset** in a dataset D if X is frequent, and there exists no super-itemset Y such that $X \subset Y$ and Y is frequent in D [7].*

3 Related Work

Many solutions have been proposed in recent years which we can categorize them into three groups:

Generate and test:
Apriori [2] or similar algorithms like [11,13,17] create frequent 1-itemsets, and then based on them build frequent 2-itemsets and the algorithm continues till finding all the frequent itemsets.

Tree based algorithms:
FP-Growth [8] or other types of algorithms [12,15] which lay in this category, draw a tree based on the items in the dataset and infer the frequent itemsets from the tree.

mmeybodi@aut.ac.ir

Hybrid:
Algorithms such as DBV-FI [16] which use combination of the two previous methods.

The various classification of frequent itemset mining algorithms are shown in Fig. 1.

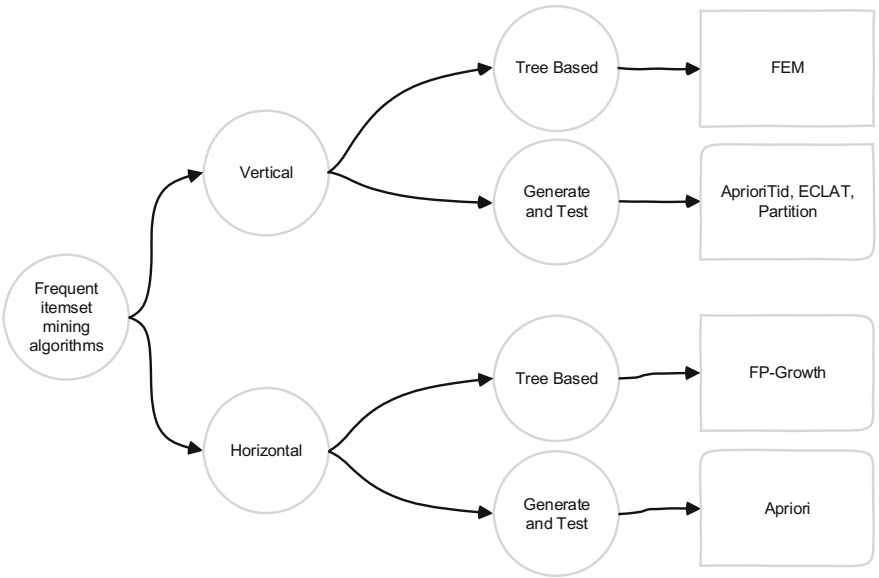


Fig. 1. Categories of different algorithms

Now let us dive deep in to the main algorithm in each category:
In this section, we are going to explain Apriori [2], Fp-Growth [8], Eclat [17]. These methods have been presented to find the Frequent Itemsets.

Apriori:
Apriori [2] is an iterative algorithm which use *generate and test* approach to find frequent Itemsets. This algorithm is a level-wise algorithm which uses the k-itemsets to find $(k+1)$-itemsets. At the first step, 1-itemsets is found by scanning whole database to accumulate the number of appearance for each item separately, then the items which can satisfy the minimum support will be collected. The resulting set is denoted by L_1 (itemsets with length1). In the next step we use L_1 in order to find L_2 (the set of frequent 2-itemsets), and we use L_2 for finding L_3 and so on. We do this procedure until no more k-itemsets can be found [7].
The time complexity of this algorithm is exponential. It may need to repeatedly scan the whole database and check a large set of candidates by pattern matching. For building L_k it needs to build all the subsets of L_k and validate

mmeybodi@aut.ac.ir

whether all the subsets satisfy the minimum support condition or not. Also finding each L_k needs to scan the whole database. Also It may still need to generate a huge number of candidate sets. For example for finding a frequent itemset of length 100 like $\{a_1, a_2, \ldots, a_{99}, a_{100}\}$, it must generate $2^{100} \approx 10^{30}$ candidates [8].

Fp-Growth:

FP-Growth [8] presents a tree-based algorithm to find frequent itemsets. The main idea of this approach is to compact the database using a tree called FP-Tree. This tree will help us to prevent from generating candidates that do not appear in the database. This will reduce the cost of searching the whole database.

The algorithm tries to scan the dataset to find 1-itemsets which satisfy the minimum support threshold, then examines only its conditional Pattern Base (a campacted database which consists of the set of frequent itemsets co-occurring with the suffix pattern) and builds a mapping from database to a tree structure so (conditional)FP-Tree will be constructed. So in the process of constructing Tree, we need to scan the transaction database twice. First finding frequent 1-itemsets, second constructing the FP-Tree. After building FP-Tree, frequent itemset mining can be performed recursively with such a tree.

The cost of inserting a transaction T in FP-tree is $O(length(T))$ where $length(T)$ means the number of frequent items in transaction T [8].

To solve the time complexity of finding frequent itemsets in recent years a lot of algorithms have been proposed.

ECLAT:

Mining frequent itemsets using the vertical data format (ECLAT) [17] algorithm improves Apriori approach by preventing from keeping lots of itemsets in memory. ECLAT uses a Vertical Database Representation. A vertical database representation indicates a list of transactions per each itemset. These kind of databases has 2 advantages.

First we can calculate the support of an itemset X by calculating the length of related set, in other words $sup(X) = |TID(X)|$, second for any itemset X and Y, the TID-list of the itemset $X \cup Y$ can be obtained without scanning the original database by intersecting the TID-lists of $X and$Y, which is $Tid(X \cup Y) = Tid(X) \cap Tid(Y)$.

ECLAT is generally faster than Apriori, but it has two disadvantages. First ECLAT also generates candidates without scanning the database, it can spend time considering itemsets that do not exist in the database. Second TID-lists can consume a lot of memory in cases that dataset is dense [5,17].

However, ECLAT method is good for small number of transactions but if the number of transactions increase this method would be inefficient. To solve this issue Deng et al. presented PPV [4] algorithm which use Node-lists data structure which is obtained from a coding prefix-tree called PPC-tree.

The comparison of the some important frequent itemsets mining algorithms can be reached in Table 1.

mmeybodi@aut.ac.ir

Table 1. Comparative study of algorithms [3,9,10]

Algorithm	Advantages	Disadvantages
AIS	• An estimation is used in the algorithm to prune those candidate itemsets that have no hope to be large. • It is suitable for low cardinality sparse transaction database.	• It is limited to only one item in the consequent. • Requires Multiple passes over the database. • Data structures required for maintaining large and candidate itemsets is not specified.
Apriori	• This algorithm has least memory consumption. • Easy implementation. • It uses Apriori property for pruning therefore, itemsets left for further support checking remain less.	• It requires many scans of database. • It allows only a single minimum support threshold. • It is favorable only for small database. • It explains only the presence or absence of an item in the database.
FP-Growth	• It is faster than other association rule mining algorithm. • It uses compressed representation of original database. • Repeated database scan is eliminated.	• The memory consumption is more. • It cannot be used for interactive mining and incremental mining. • The resulting FP-Tree is not unique for the same logical database
MAXCONF	• Using the two pruning methods reduce the row enumeration space. • It mines all common relationships and rare interesting relationships. • It is faster than algorithms such as APRIORI, MAX-MINER.	• The intersection amongst the itemsets leads to time and space consumption. • It still produces non-maximal rules.
ECLAT	• Scanning the database to find the support count of (k + 1)-itemsets is not required.	• More memory space and processing time are required for intersecting long TID sets.

4 Proposed Algorithm

In this section we expound how our algorithm works. Clustering techniques could be used in the fields of data mining in order to reduce the size of the data. What we achieve by clustering our data set is some clusters which consist of itemsets that used in similar transactions. Therefore there is a high probability that each of the frequent itemsets become a subset of one of our clusters.

In most algorithm which represented in recent years finding frequent itemsets are a bottom-up approach it means that algorithms first find 1-itemsets, then find 2-itemsets and so on. Nevertheless what we presented here is not a bottom-up approach. Our proposed algorithm tends to find longest frequent itemsets as a result in most cases it finds maximal frequent itemsets.

mmeybodi@aut.ac.ir

The idea of partitioning the data in frequent-pattern mining was introduced before, in partition-based approach, the transactions are divided into K partitions, and the frequent pattern mining algorithm is applied on each of these partitions. Then, the results of different partitions are integrated to build the final frequent pattern list. In this approach, the final results, should be investigated to be frequent considering the total transactions in database.

The main idea behind our proposed approach in that if we cluster the transactions instead of random partitioning, we hope to reach to better results. In other word, we use smart partitioning approach, which use clustering to find similar transactions and put them in one partition. Now what we yearn is either our clusters are frequent or not. Therefore we just need to count the frequency of each cluster, so some how our algorithm change the problem of frequent mining to search for number of existence of a pattern problem. What we suggested in this paper for this part is just a simple search which can be improved.

Building an efficient frequent mining algorithm that can handle big data, has efficient time complexity and has acceptable accuracy, needs to overcome several challenges:

Challenge 1: Data Representation. Either we have a binary representation of our dataset or we should make one. To make the dataset first we should transform each item to its one-hot encoding representation. As a result each transaction would be equal to a vector which consist of 1s if the item exist and 0s if item does not exist.

Challenge 2: Efficient Clustering. Now we should cluster our dataset. We examined several clustering algorithms in this era. Applying clustering to find similar transactions in large datasets, needs an efficient clustering algorithm, in other word, clustering is a bottleneck in clustering-based frequent pattern mining algorithm. We examine some of the clustering algorithms such as k-means and finally we find out that an incremental clustering approach can be useful in large scale data, so we employ mini batch k-means [14] clustering technique. Pseudo-code of the mini-batch clustering technique can be reached in Algorithm 1. The input of our proposed algorithm is a Dataset (D), the number of clusters (K), a Mini-Batch Size (b), and the number of iterations (t) as input.

Challenge 3: Find the Representative of a Cluster. When the clustering is done, we have k clusters. Each cluster has a vector of numbers, each number in the vector is between 0 and 1. Let's define $c[i]$, $c[i]$ is the corresponding number to item i in a cluster c, $c[i]$ shows how much the item i is dependent to cluster c. If $c[i]$ is close to 0 it means item i is not dependent to cluster c, and if $c[i]$ is close to 1 it means item i is dependent to cluster c.

We need to smooth the $c[i]$ in each cluster c, we define a threshold $\theta = 0.5$. We have chosen 0.5 because it means half of the cluster members have item i, it is a good condition to check whether more than half of the cluster members have item i or not. If $c[i]$ is strictly greater than 0.5, then we put $c[i] = 1$, otherwise if $c[i]$ is less than or equal to 0.5, then $c[i] = 0$. For example if $c[i] = 0.3$ it means 30% of cluster members have item i and 70% of cluster members doesn't have

mmeybodi@aut.ac.ir

Algorithm 1. Mini-batch K-means [14]

Input : k, mini-batch size b, iterations t, dataset D
Initialize each $c \in C$ with an x picked randomly from D
$v \leftarrow 0$
for $i = 1$ to t **do**
  $M \leftarrow b$ examples picked randomly from D
  for $x \in M$ **do**
    $d[x] \leftarrow f(C, x)$ ▷ Cache the center nearest to x
  end for
  for $X \in M$ **do**
    $c \leftarrow d[x]$ ▷ Get cached center for this x
    $v[c] \leftarrow v[c] + 1$ ▷ Update per-center counts
    $\eta \leftarrow \frac{1}{v[c]}$ ▷ Get per-center learning rate
    $c \leftarrow (1 - \eta)c + \eta x$ ▷ Take gradient step
  end for
end for

item i. After running Mini-Batch K-Means and smoothing, each cluster can be a candidate for being a frequent itemset.

Challenge 4: Pruning. For validating these candidates, we need to define a min-support as a threshold, then we'll iterate over the dataset to check whether the candidates have enough support or not, in other words the itemset is it frequent or not. So we need to scan dataset once to check the validity of our frequent itemsets.

The pseudo-code of our proposed algorithm presented in Algorithm 2.

5 Experiments

The dataset which we used in this experiment can be reached form GitHub[1]. First we start with a dataset of 5000 transactions and in each step we increase the size of the dataset by 1000 transactions. The maximum size of the dataset has 75000 transactions and the maximum length of the transactions is equal to 8 and the database has 50 items. The selected minimum support for the experiment is $0.008 \times \mathsf{number\ of\ transactions}$. For example, if dataset has 5000 transactions the minimum support would be equal to 40. It would be an obvious fact if the number of transactions increase the minimum support would be increase accordingly.

In our proposed algorithm we chosen 150 cluster and the batch size is equal to 200 and the number of iteration is equal to 20. In each step we compute time consumption of our proposed algorithm and compare it to FP-Growth algorithm. To be assured of the performance of the proposed algorithm we run the experiment 10 times and the result is available in Fig. 2.

Note that we implemented our code in python 3 and the hardware which we used has a *core i7* CPU and *8 Gigabyte* of RAM.

[1] https://github.com/timothyasp/apriori-python.

mmeybodi@aut.ac.ir

Algorithm 2. Proposed Algorithm

Input : k, mini batch size b, number of iterations t, dataset X, min-support-ratio θ

run mini batch k-means on dataset X and find Cluster centers c

min-support $= \theta \times length(X)$

// Smoothing the Clusters
for each cluster **do**
  round each number in cluster
end for
frequent-itemset = empty list
for each cluster **do**
  $candidate$ = itemset made from cluster
  $cluster - support$ = the number of appearance for cluster in dataset X
  if $cluster - support \geq$ min-support **then**
    Add candidate to frequent-itemset
  end if
end for

Top 10 clusters deduced from dataset with respect to number of transactions and number of occurrence of each cluster and minimum support is presented in Table 2.

We have 150 clusters which contain almost all of the frequent itemsets. Nevertheless it has some clusters which are not frequent. The number of frequent clusters in respect to all of the clusters has depict in the Fig. 3.

By counting the frequency of each cluster we can remove the non-frequent clusters but It would be time consuming and it can decrease the accuracy of the algorithm. Figure 4 shows time consumption of our algorithm in addition to counting frequency of each cluster in contrast to FP-Growth and Fig. 5 shows the accuracy of it.

If we use curve fitting methods to find an equation for FP-Growth algorithm based on our inferred result from our experiment, we will get the following equation:

$$\begin{aligned} Time = 2 \times 10^{-9} \times \#transaction^2 + 2 \times 10^{-5} \times \#transaction - 0.0272 \\ \text{the coefficient of determination: } R^2 = 0.9999 \end{aligned} \tag{2}$$

If we use curve fitting methods to find an equation for the proposed algorithm based on our inferred result from our experiment, we will get the following equation:

$$\begin{aligned} Time = 7 \times 10^{-5} \times \#transaction + 0.2933 \\ \text{the coefficient of determination: } R^2 = 0.9982 \end{aligned} \tag{3}$$

mmeybodi@aut.ac.ir

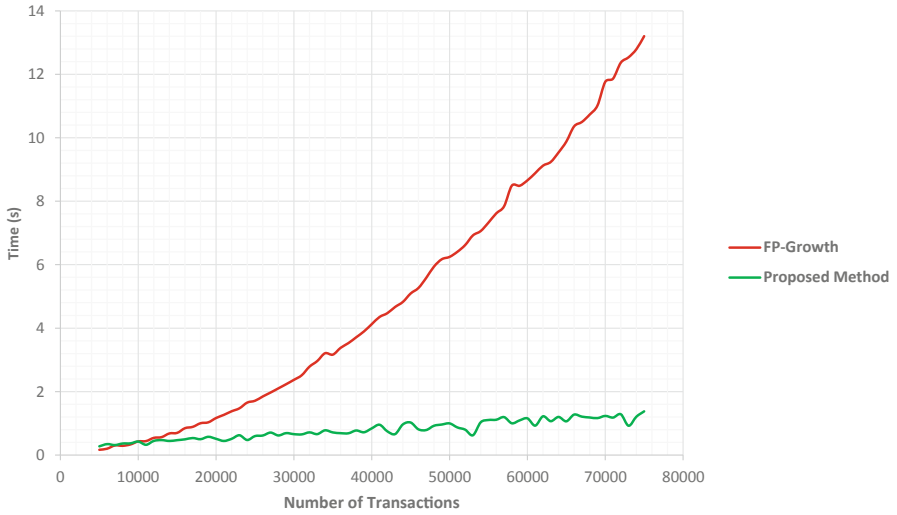


Fig. 2. Time consumption of our proposed algorithm vs FP-Growth

Table 2. Top 10 cluster frequency

Top 10 Clusters	Number of Transactions in Cluster	Candidate ItemSet Made from cluster	Support-count Itemset in dataset	Min-Support
1	3614	{ 27 , 28 }	3819	600
2	2983	{ 3 , 18 , 35 }	3083	600
3	2660	{ 1 , 19 }	2764	600
4	2571	{ 17 , 29 , 47 }	2007	600
5	2098	{ 7 , 15 , 49 }	2040	600
6	2000	{ 0 , 2 , 46 }	2504	600
7	1945	{ 12 , 31 , 36 , 48 }	1544	600
8	1858	{ 7 , 11 , 37 , 45 }	2094	600
9	1761	{ 16 , 32 , 45 }	2462	600
10	1651	{ 1 }	6271	600

mmeybodi@aut.ac.ir



Fig. 3. Ratio of frequent clusters to all of the constructed clusters

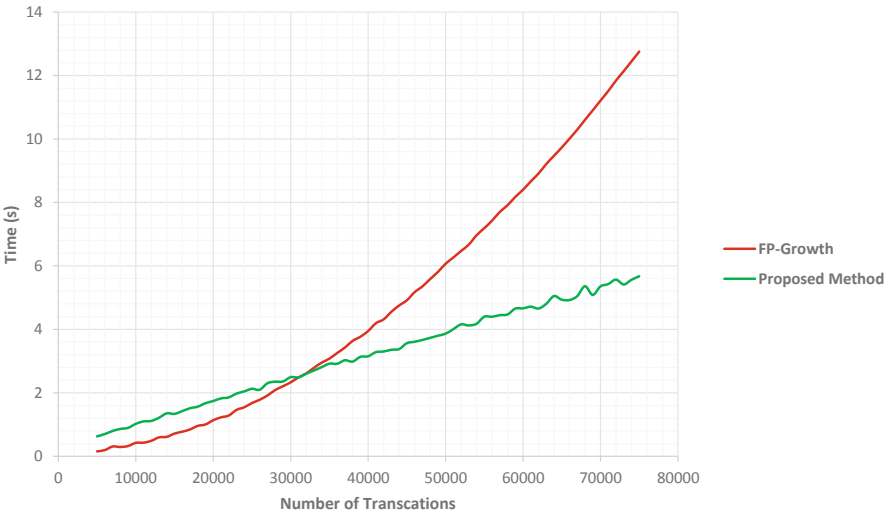


Fig. 4. Time consumption of our proposed algorithm with checking for frequent clusters vs FP-Growth

mmeybodi@aut.ac.ir



Fig. 5. Accuracy of proposed algorithm with checking for frequent clusters

6 Conclusion

In this paper, we introduced an efficient approximate algorithm to mine frequent itemsets in a set of transactions. To find frequent patterns, we first represent each transaction by a binary vector where the i-th entry is 1 if the i-th item is present in the transaction. We then use an approximate version of K-means clustering, called mini-batch K-means, to group similar transactions together. The center of induced clusters are considered as potential frequent itemsets. To further test this assumption, we count the support of each cluster center. Experiments show that the execution time of our presented algorithm is linear. Moreover, our proposed algorithm has proved to be more performant than FP-Growth algorithm on various databases.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. SIGMOD Rec. **22**(2), 207–216 (1993). https://doi.org/10.1145/170036.170072
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB 1994, pp. 487–499. Morgan Kaufmann Publishers Inc., San Francisco (1994). http://dl.acm.org/citation.cfm?id=645920.672836
3. Bayardo Jr, R.J.: Efficiently mining long patterns from databases. In: ACM SIGMOD Record, vol. 27, pp. 85–93. ACM (1998)
4. Deng, Z., Wang, Z.: A new fast vertical method for mining frequent patterns. Int. J. Comput. Intell. Syst. **3**, 733–744 (2010). https://doi.org/10.2991/ijcis.2010.3.6.4

mmeybodi@aut.ac.ir

5. Fournier-Viger, P., Lin, J.C.W., Vo, B., Chi, T.T., Zhang, J., Le, H.B.: A survey of itemset mining. Wiley Interdisc. Rev.: Data Min. Knowl. Discovery **7**(4), e1207 (2017)
6. Hahsler, M., Grün, B., Hornik, K., Buchta, C.: Introduction to arules – a computational environment for mining association rules and frequent item sets (2005)
7. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco (2011)
8. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. SIGMOD Rec. **29**(2), 1–12 (2000). https://doi.org/10.1145/335191.335372
9. Kaur, J., Madan, N.: Association rule mining: a survey. Int. J. Hybrid Inf. Technol. **8**(7), 239–242 (2015)
10. McIntosh, T., Chawla, S.: High confidence rule mining for microarray analysis. IEEE/ACM Trans. Comput. Biol. Bioinf. (TCBB) **4**(4), 611–623 (2007)
11. Park, J.S., Chen, M.S., Yu, P.S.: An effective hash-based algorithm for mining association rules. SIGMOD Rec. **24**(2), 175–186 (1995). https://doi.org/10.1145/568271.223813
12. Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., Yang, D.: H-mine: hyper-structure mining of frequent patterns in large databases. In: Proceedings 2001 IEEE International Conference on Data Mining, pp. 441–448, November 2001. https://doi.org/10.1109/ICDM.2001.989550
13. Savasere, A., Omiecinski, E., Navathe, S.B.: An efficient algorithm for mining association rules in large databases. In: Proceedings of the 21th International Conference on Very Large Data Bases, VLDB 1995, pp. 432–444. Morgan Kaufmann Publishers Inc., San Francisco (1995). http://dl.acm.org/citation.cfm?id=645921.673300
14. Sculley, D.: Web-scale k-means clustering. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 1177–1178. ACM, New York (2010). https://doi.org/10.1145/1772690.1772862
15. Uno, T., Kiyomi, M., Arimura, H.: Efficient mining algorithms for frequent/closed/maximal itemsets. In: Proceedings of the IEEE ICDM Workshop Frequent Itemset Mining Implementations (2004)
16. Vo, B., Hong, T.P., Le, B.: Dynamic bit vectors: an efficient approach for mining frequent itemsets. Sci. Res. Essays **6**(25), 5358–5368 (2011)
17. Zaki, M.J.: Scalable algorithms for association mining. IEEE Trans. Knowl. Data Eng. **12**(3), 372–390 (2000). https://doi.org/10.1109/69.846291

mmeybodi@aut.ac.ir



Next Frame Prediction Using Flow Fields

Roghayeh Pazoki and Parvin Razzaghi(✉)

Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran
{r.pazoki,p.razzaghi}@iasbs.ac.ir

Abstract. Next frame prediction is the challenging task in computer vision and video prediction. Despite the longtime studies in video processing, the next frame prediction problem is rarely investigated and it is at its beginning. In next frame prediction, the main goal is to design a model which automatically generates the next frame using a sequence of previous frames. In videos, in most cases, the large portion of the current frame is similar to the previous frames and only a small portion of the frame has a motion field. This leads us to utilize the optic flow field. To do so, Laplacian pyramid of convolutional networks and adversarial learning are used to predict simultaneously the optic flow and the gray content of the next frame. To evaluate the proposed approach, it is applied on UCF101 dataset. The obtained results show that our approach achieves a better performance.

Keywords: Frame prediction · Generative adversarial networks · Optic flow

1 Introduction

Next frame prediction in videos is a challenging problem in computer vision which has been received interest in the recent years. It has several real-world applications in robotics [9, 10], prediction of abnormal situations in surveillance and human action prediction.

One of the major challenges which should be considered in the video prediction is the uncertainty of the future and the nature of its multimodality. Vondrick et al. [21] proposed a convolutional neural network to predict the visual representation of future frames. Then, they applied recognition algorithm on the predicted representation to predict human actions and objects in the future. Their proposed network is pretrained using a large amount of unlabeled videos. In [22], Vondrick et al. explored the problem of learning how scenes transform with time. To do this, a model is proposed to learn scene dynamics for video generation and video recognition tasks, using a large amount of unlabeled videos. Oh et al. [15] proposed two different deep architectures as an action conditional auto-encoder to predict long term next frame sequences in Atari games. Lotter et al. [11] defined a recurrent convolutional network, inspired by the concept of predictive coding from the neuroscience literature, to continually predict the appearance of future frames. Srivastava et al. [20] used a Long Short-Term Memory (LSTM) network [18] to learn the representation of video sequences in an unsupervised manner and then utilized it to predict the future frames. Ranzato et al. [17] utilized a recurrent network architecture, inspired by language modeling, to predict the frames in

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 238–247, 2020.
https://doi.org/10.1007/978-3-030-37309-2_19

mmeybodi@aut.ac.ir

a discrete space of patch clusters. In the mentioned works [17, 20], the uncertainty of the future is not considered, hence, the blur effect is observed mainly at the predicted frames. Mathieu et al. [13] proposed an approach to consider this issue. To do this, they utilized multi-scale architecture along with adversarial generative loss [5] and image gradient difference loss function to cope this challenge.

Up to now, in next frame prediction, the pixels values of the whole entire frame have been predicted. However, consecutive frames in the video are often very similar to each other, and usually, the background is fixed and only parts of the image have movements. When a man wants to predict the next frame, he usually concentrates on the moving parts of the current frame. To consider this issue, we incorporate the optic flow of the previous frames in next frame prediction. In this paper, in order to simultaneously predict the appearance and optic flow of the next frame, the multi-scale deep convolutional generative network along with the adversarial learning are utilized.

This paper is organized as follows: Sect. 2 describes the whole proposed approach. The contribution of the proposed approach is given in Sect. 2.2. The experimental results are given in Sect. 3. Finally, in Sect. 4, the paper is concluded.

2 Approach

Let $x = \{x^1, x^2, \ldots, x^m\}$ be a sequence of input frames where x^i denotes the i^{th} input frame. The goal is to predict the next frame x^{m+1} which is denoted by y in the rest of the paper. As it is stated, the consecutive frames are very similar to each other, and only some parts of the frame have movements. In the proposed architecture, both the appearance of the next frame and its optic flow are predicted. In the inference step, the next frame is obtained by warping the current frame with the predicted optical flow. In the following, each step of the proposed approach is explained in detail.

2.1 Model

In this section, we present a model for next frame prediction. Similar to [13], we use a combination of Laplacian pyramid of convolutional networks and adversarial learning [3], which shows good performance on the next frame prediction.

Generative adversarial models [5] consist of two networks, generator G and discriminator D, which are trained competitively. The generator is trained to produce a similar image to real data from random noise which is indistinguishable from the real image for network D, and discriminator D is trained to distinguish the generated image by G. Training these networks are done simultaneously.

In the next frame prediction [13], the generator G is trained to predict the next frame y of input sequence frames $x = \{x^1, \ldots, x^m\}$ and the discriminator D takes a sequence of frames as input in which all frames except the last ones are from the dataset. The last frames can be from the dataset or are generated by G. The discriminator network D is trained to predict whether the last frame is real or predicted by the generator network G. In the following, a unified framework is given which explains how Laplacian pyramid is combined with adversarial network.

mmeybodi@aut.ac.ir

This framework contains N different generator networks in N different scales, which are shown by $G = \{G_1, \ldots, G_N\}$ and their corresponding discriminator networks are shown by $D = \{D_1, \ldots, D_N\}$. Each G_k is a convolutional network, which takes x_k in scale k and the up-sampled predicted frame which is generated by G_{k-1} as input and it predicts $y_k - u_k(y_{k-1})$ based on Laplacian pyramid approach [1]. In other words, G_k predicts $\widehat{y}_k$ as follows:

$$\widehat{y}_k = u_k(\widehat{y}_{k-1}) + G_k(x_k, u_k(\widehat{y}_{k-1})), \tag{1}$$

where u_k denotes the up-sampling function.

In this framework, prediction starts from the lowest scale and the generative model G predicts a series of the next frame in N scales. In other words, the predicted result in the k^{th} scale is utilized to predict the result, in the $k+1^{th}$ scale. This framework gradually leads the approach toward full resolution prediction. In Fig. 1, the scheme of the proposed approach is shown.

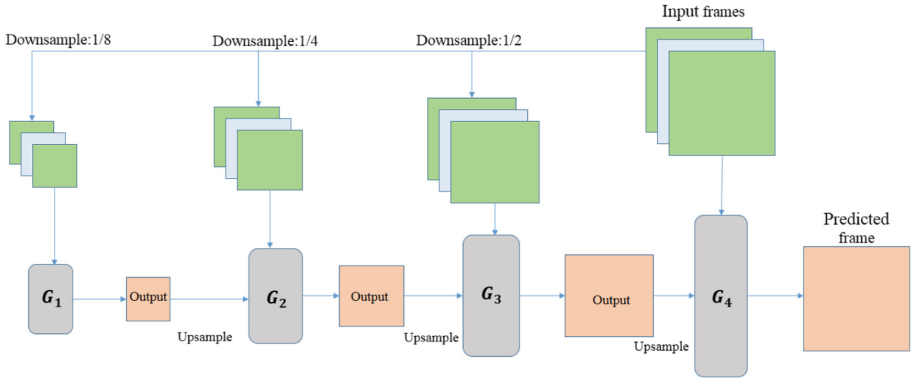


Fig. 1. The scheme of the multiscale generative model in four scales. Prediction starts from the lowest scale.

2.2 Optic Flow Integration

In this subsection, our contribution which explains how optic flow is incorporated in next frame prediction is given. In [13], the multi-scale generative model is trained using a sequence of frames and produces the whole next frame. As it is stated, the consecutive frames are similar and only some parts of these frames are different. Hence, the whole frame prediction causes a blur in the static parts of the frame. Therefore, we utilize the optic flow of each frame to predict the optic flow of the next frame. In other words, rather than directly predicting the next frame, we predict the optical flow of next frame and then warp the predicted optical flow with the last input frame to construct the next frame.

The optical flow [7, 12] represents the motion of pixels between two consecutive input frames. The optical flow of each frame is sparse, so it causes only some part of the frame to have flow vector. It should be noted that the correct optical flow is not

mmeybodi@aut.ac.ir

available for real-world videos. There are many works which compute the optical flow field between two consecutive images. In this paper, to compute the optical flow field and to feed it as ground truth flow field into the proposed network, SpyNet method [16] is utilized. SpyNet [16] is an optical flow method based on a combination of classical optic flow algorithm and deep learning. It uses a spatial pyramid structure in which each level contains convolutional networks which are trained to estimate a flow update at each level and to compute optical flow in a coarse-to-fine way.

Since we extract the optical flow of each frame by SpyNet [16] and use it as ground truth optic flows in the proposed approach, the error in these flow fields are propagated in the whole approach. As a result, to reduce this error in the whole network, we simultaneously predict the optic flow and the gray scale of the next frame. To do this, we concatenate the grayscale images of the input frames with their optic flows, such that, the input sequence in our approach will contain two dimensions for optic flow and one more for the grayscale image of the second frames (see Fig. 2).

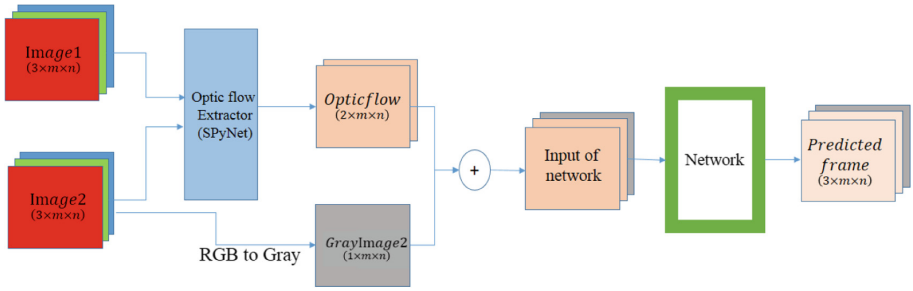


Fig. 2. The scheme of how optic flow field and the appearance information is combined to provide the input of the proposed approach.

The training procedure of the model is explained in the following section.

2.3 Model Training

The multi-scale model is trained through the combination of the reconstruction loss and the adversarial loss. This model is trained by optimizing the following minimax objective:

$$\min_G \max_D \lambda_p \mathcal{L}_p(G) + \lambda_{adv} \mathcal{L}_{adv}(G, D), \tag{2}$$

where λ_{adv} and λ_p respectively control the importance of the adversarial loss and the reconstruction loss in model training. Training the generator network and the discriminator network are done respectively. In other words, the discriminator is trained while the generator is fixed and then the generator is trained while the discriminator is fixed. This procedure is repeatedly done until convergence is reached.

mmeybodi@aut.ac.ir

Training Discriminator D. Model D is trained to discriminate the true next frame from the generated one. To do so, two classes are considered. Discriminator network D should classify y as belonging to class 1 and the generated optic flow and grayscale image as belonging to class 0. To do this, one can use binary cross-entropy loss which is defined as:

$$\mathcal{L}_{bce}(p, l) = -l \log(p) - (1 - l) \log(1 - p), \tag{3}$$

where p is the output probability of the discriminator network that is in [0, 1] interval and l is the class label of data that is in {0, 1}. Minimizing the cross-entropy loss is equivalent to maximizing the adversarial loss. Hence, the adversarial loss for training network D is defined as:

$$\mathcal{L}^{D}_{adv}(x, y) = \mathcal{L}_{bce}(D(x, y), 1) + \mathcal{L}_{bce}(D(x, G(x)), 0). \tag{4}$$

As mentioned, in the proposed method, the multi-scale architecture is used. As a result, the discriminative network D minimizes the following objective loss:

$$\mathcal{L}^{D}_{adv}(x, y) = \sum\nolimits_{k=1}^{N} \mathcal{L}_{bce}(D_k(x_k, y_k), 1) + \mathcal{L}_{bce}(D_k(x_k, G_k(x_k, \widehat{y}_{k-1})), 0). \tag{5}$$

This loss function is minimized when (x_k, y_k) is classified as a real frame (class 1) and the generated frame $(x_k, G_k(x_k, \widehat{y}_{k-1}))$ is classified as a false one (class 0).

Training Generator G. The generator G tries to generate the next frame such that D cannot distinguish the generated next frame with the real next frame. In order to train generator G, by fixing the parameters of discriminator D, the following objective function is minimized:

$$\mathcal{L}^{G}(x, y) = \lambda_{adv} \mathcal{L}^{G}_{adv}(x, y) + \lambda_{p} \mathcal{L}_{p}(x, y), \tag{6}$$

where $\mathcal{L}^{G}_{adv}$ denotes the adversarial loss of the network G and $\mathcal{L}_{p}$ denotes the reconstruction loss. In the following, all of these loss functions are defined in detail.

In this paper, to define $\mathcal{L}^{G}_{adv}$, similar to [13], the following function is utilized:

$$\sum_{k=1}^{N} \mathcal{L}_{bce}(D_k(x_k, G_k(x_k, \widehat{y}_{k-1})), 1), \tag{7}$$

where k denotes the scale index of the generator and discriminator networks in the multi-scale architecture. This loss function is minimized when the discriminator of each scale classifies the generated frame as the real one.

mmeybodi@aut.ac.ir

The reconstruction loss in this paper is defined as follows:

$$\mathcal{L}_p(x,y) = \lambda_{opt}\left\|opt_{\hat{y}} - opt_y\right\|_p^p + \lambda_{gray}\left\|gray_{\hat{y}} - gray_y\right\|_p^p, \tag{8}$$

where the first term minimizes the distance between the predicted optic flow $opt_{\hat{y}}$ and the true optic flow opt_y, the second term minimizes the distance between the predicted grayscale image $gray_{\hat{y}}$ and the true grayscale image $gray_y$. Also, λ_{opt} and λ_{gray} are the control variables.

3 Experiments

In this section, the proposed model is evaluated. In so doing, the model is applied on UCF101 dataset [19]. The UCF101 dataset contains 13320 videos, which belong to 101 classes of human actions. This dataset is divided into two disjoint training and test sets, which contain 9500 and 3820 videos respectively. Each video has a different length and the resolution of each frame is 240 $\times$ 320. To train the proposed model, the sequences of patches of size 32 $\times$ 32 pixels, which have enough motion, are sampled, similar to [13]. First, we normalize the sequences and determine the optical flow of the successive frames by SpyNet [16], and then we normalize their value to $[-1, 1]$ interval. The extracted optic flow of two successive frames is concatenated with the grayscale of the second frame and is fed as input to the proposed model. It should be noted that to predict more than one frame, the model is recursively applied on the newly generated frame as an input.

The model is implemented in Torch7 [2]. Training is done on a system with Nvidia Geforce GTX 960 GPU. In the training phase, the learning rate and the batch size, respectively are set to 0.02 and 8; and the optimization is done via Stochastic Gradient Descent (SGD) algorithm.

3.1 Model Architecture

Similar to [13], our proposed model has four scale levels: $s_1 = 4 \times 4$, $s_2 = 8 \times 8$, $s_3 = 16 \times 16$ and $s_4 = 32 \times 32$, where each level contains a generator and a discriminator. The architecture of the model is shown in Table 1. The generative model is a fully convolutional network, which consists of padded convolution layers followed by rectified linear units (ReLU) [14]. There is a hyperbolic tangent layer at the end of the model in order to ensure that the outputs are in range $[-1, 1]$.

Table 1. The multi-scale network architecture used in four scales

Generative networks	G1	G2	G3	G4
Conv. kernel size	3, 3, 3, 3	5, 3, 3, 5	5, 3, 3, 3, 3, 5	7, 5, 5, 5, 5, 7
#Feature maps	16, 32, 16	16, 32, 16	16, 32, 64, 32, 16	16, 32, 64, 32, 16
Discriminative networks	D1	D2	D3	D4
Conv. kernel size	3	3, 3, 3	5, 5, 5	7, 7, 5, 5
#Feature maps	32	32, 32, 64	32, 32, 64	32, 32, 64, 128
Fully connected	256, 128	256, 128	256, 128	256, 128

mmeybodi@aut.ac.ir

The discriminative model contains convolution layers followed by the rectified linear units and fully connected layers. For D_4, a 2×2 pooling layer is added after the convolution layers.

3.2 Quality Metrics

To evaluate the quality of the reconstructed frame, similar to [4, 6, 13, 15], we use Peak Signal-to-Noise Ratio (PSNR), sharpness difference and Structural Similarity Index Measure (SSIM) [23] as similarity measures. In the following, these similarity measures are defined.

PSNR measure is defined as follows:

$$PSNR(y, \hat{y}) = 10 \log_{10} \frac{max_{\hat{y}}^2}{\frac{1}{N} \sum_{i=0}^{N} (y_i, \hat{y}_i)^2}, \tag{9}$$

where y and $\hat{y}$ are the true frame and the generated frame respectively, and $max_{\hat{y}}$ is the maximum possible intensity of the image.

Sharpness difference [13] measures the loss of sharpness between the generated frame and the true frame. It is based on the difference of gradients between the two images namely y and $\hat{y}$:

$$Sharp.diff(y, \hat{y}) = 10 \log_{10} \frac{max_{\hat{y}}^2}{\frac{1}{N} \left(\sum_i \sum_j \left| \left(\nabla_i y + \nabla_j y \right) - \left(\nabla_i \hat{y} + \nabla_j \hat{y} \right) \right| \right)}, \tag{10}$$

where $\nabla_i y = \left| y_{i,j} - y_{i-1,j} \right|$ and $\nabla_j y = \left| y_{i,j} - y_{i,j-1} \right|$.

Another metric is SSIM, whose value is in range [0, 1], where the larger value admits high similarity between two images.

3.3 Results

In order to evaluate the performance of the proposed model, similar to the comparable approaches, we apply the trained model on a subset of UCF101 test dataset [19], which contains 379 videos and measure the quality of the generated image by the predicted optic flow via the mentioned metrics.

The model is trained using different values of the control parameters for the adversarial loss and effectiveness of the gray images in the reconstruction loss. In all experiments, we have set $p = 2$ in the reconstruction loss and the weight of it is set to 1 similar to [13]. The optic flow control parameter λ_{opt} in the reconstruction loss is computed by $\lambda_{gray} + \lambda_{opt} = 1$. Table 2 represents the quantitative evaluation between next target frame and next reconstructed frame. When λ_{adv} is set to 0.05, we get better results compared to the time when it is set to other values. Using larger or smaller values of λ_{adv} may decrease the performance. Therefore, we choose $\lambda_{adv} = 0.05$, then we adjust λ_{gray} in range [0.2, 0.8] by step size 0.2 to validate the effect of the gray images in training. The results show that the model reaches the best performance in $\lambda_{gray} = 0.4$.

mmeybodi@aut.ac.ir

Table 2. The obtained results of the proposed approach on UCF101. The proposed approach is evaluated on different values of λ_{adv} and λ_{gray}.

Parameters		1st frame prediction scores			2nd frame prediction scores		
λ_{adv}	λ_{gray}	PSNR	SSIM	Sharpness	PSNR	SSIM	Sharpness
0.01	0.8	18.44	0.61	16.59	16.01	0.54	16.20
0.01	0.6	19.41	0.67	17.13	17.10	0.57	16.78
0.05	0.8	20.61	0.75	17.40	18.55	0.67	16.85
0.05	**0.4**	**26.97**	**0.89**	**19.70**	**23.45**	**0.82**	**18.56**
0.05	0.2	25.80	0.87	19.37	22.58	0.80	18.37
0.07	0.2	25.52	0.87	19.20	22.26	0.79	18.22

In Table 3, the proposed model is compared with the base approaches and [13]. In [13], the model is trained using Sport1m dataset [8], which contains 1 million sport video clips from YouTube. Then their the best model has been fine-tuned by the patches of size 64 × 64 on the UCF101 dataset [19], after the training on the Sport1m dataset (Our model is trained only on the UCF101 dataset.).

In Table 3, $\mathcal{L}_2$ and GDL+$\mathcal{L}_1$ present the results for their model, which have been trained respectively using $\mathcal{L}_2$ loss and a combination of the gradient difference loss and the $\mathcal{L}_1$ loss. Also, Adv and Adv+GDL, have been trained using, adversarial loss with the $\mathcal{L}_2$ loss and a combination of the adversarial loss and the gradient difference loss, respectively.

As shown in Table 3, our approach in SSIM and Sharpness receives better results compared to the other approach. Also, in PSNR, our approach obtains a comparable result compared to Adv+GDL approach. In the second predicted frame, our approach in all measures gets better results compared to the other approach. These results confirm that the incorporation of the optic flow in next frame prediction leads to an increase in performance. As stated, there is not any ground truth optical flow for real-world videos, so we train our model using the extracted optic flow by the SpyNet [16] as ground truth next optical flow. Nevertheless, the obtained results are satisfying and our proposed approach is successful in maintaining the static portions nearly intact.

Table 3. The comparison of the proposed approach with the base approaches and the different version of approach [13].

Approach	1st frame prediction scores			2nd frame prediction scores		
	PSNR	SSIM	Sharpness	PSNR	SSIM	Sharpness
Ours	26.97	**0.89**	**19.70**	**23.45**	**0.82**	**18.56**
$\mathcal{L}_2$	20.10	0.64	17.80	14.10	0.50	17.40
GDL+$\mathcal{L}_1$	23.90	0.80	18.70	18.60	0.64	17.70
Adv	24.16	0.76	18.64	18.80	0.59	17.25
Adv+GDL	**27.06**	0.83	19.54	22.55	0.71	18.49

mmeybodi@aut.ac.ir

4 Conclusion

In this paper, a new approach for next frame prediction is proposed. To do so, the multi-scale generative model is presented which can simultaneously predict the appearance and the optic flow of the next frame. It causes that the proposed approach only concentrates on the moving parts of the frame. To evaluate the proposed approach, it is applied on UCF101 dataset and the obtained results show that the proposed approach does better than the comparable approaches. In the future work, one can examine how the layer-wise optical flow impact on the next frame prediction.

References

1. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image code. In: Readings in Computer Vision, pp. 671–679. Elsevier (1987)
2. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: a Matlab-like environment for machine learning. In: BigLearn, NIPS Workshop, No. EPFL-CONF-192376 (2011)
3. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a Laplacian pyramid of adversarial networks. In: Advances in Neural Information Processing Systems, pp. 1486–1494 (2015)
4. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: Advances in Neural Information Processing Systems, pp. 64–72 (2016)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing systems, pp. 2672–2680 (2014)
6. Hore, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 2366–2369. IEEE (2010)
7. Horn, B.K., Schunck, B.G.: Determining optical flow. Artif. Intell. **17**(1–3), 185–203 (1981)
8. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)
9. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. IEEE Trans. Pattern Anal. Mach. Intell. **38**(1), 14–29 (2016)
10. Kosaka, A., Kak, A.C.: Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. CVGIP: Image Underst. **56**(3), 271–329 (1992)
11. Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint arXiv:1605.08104 (2016)
12. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision (1981)
13. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: ICLR (2016)
14. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning, ICML 2010, pp. 807–814 (2010)
15. Oh, J., Guo, X., Lee, H., Lewis, R.L., Singh, S.: Action-conditional video prediction using deep networks in Atari games. In: Advances in Neural Information Processing Systems, pp. 2863–2871 (2015)

mmeybodi@aut.ac.ir

16. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2 (2017)
17. Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., Chopra, S.: Video (language) modeling: a baseline for generative models of natural videos. arXiv preprint arXiv:1412.6604 (2014)
18. Schmidhuber, J., Hochreiter, S.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
19. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild. CoRR, abs/1212.0402 (2012)
20. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using LSTMs. In: ICML, pp. 843–852 (2015)
21. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating visual representations from unlabeled video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 98–106 (2016)
22. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Advances In Neural Information Processing Systems, pp. 613–621 (2016)
23. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)

mmeybodi@aut.ac.ir



Using Augmented Genetic Algorithm for Search-Based Software Testing

Zahir Hasheminasab, Zaniar Sharifi(✉), Khabat Soltanian, and Mohsen Afsharchi

Zanjan University, Zanjan, Iran
zaniar.sharifi@znu.ac.ir

Abstract. Automatic test case generation has been received great attention by researchers. Evolutionary algorithms have increasingly gained special places as means of automating the test data generation for software testing. Genetic algorithm (GA) is the most commonplace algorithm in search-based software testing. One of the key issues of search-based testing is the inefficient and inadequate informed fitness function due to the rigidness of fitness landscape. To deal with this problem, in this paper we improved a recently published fundamental approach where a new criterion, branch hardness factor is used to calculate fitness. However, the existing methods are unable to cover the whole of the targets. Herein, we added a local search strategy to the standard GA for faster convergence and providing more intensification. In addition, different selection and mutation operators are examined and appropriate choices selected. Our approach gained remarkable efficiencies on 7 standard benchmarks. The results showed that adding local search is likely to boost another search-based algorithm for path coverage even.

Keywords: Genetic algorithm · Path coverage testing · Automatic test data generation

1 Introduction

Nowadays, using software production is becoming more and more indispensable in daily life, therefore role of software testing is being highlighted for verifying quality of software. Approximately 50 percent of software development process cost is being consumed on Software testing [1]. Moreover, this process is a time consuming and tedious process, since it is done manually. Therefore, automated software testing is being evaluated as the indispensable method to decline time and cost.

There are different types of testing criteria, which are classified into two testing strategies, such as black-box testing and white-box testing [2]. Black box testing, is a software testing method which the code block being tested is not known, whereas white-box testing is respected to only the implementation of items that can be tested.

In the other word, in white box testing the internal structure of the program under test is known for tester.

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 248–257, 2020.
https://doi.org/10.1007/978-3-030-37309-2_20
mmeybodi@aut.ac.ir

Generally Speaking, the main goal of software testing is to generate test cases satisfying test criteria. The test cases are sets of terms or variables that testers will test them to determine whether a system under test satisfied conditions.

Test case generation approaches based on the algorithm can be classified to static methods, dynamic methods and hybrid methods.

Static methods are software testing techniques in which the software is tested without executing the code. They comprise symbolic execution [4] and domain reduction [5, 6]. Although these methods have had important successes, they still face challenges in managing procedure calls, indefinite loops, pointer references and array in any tested program [7].

In symbolic execution method, instead of using actual value, symbolic value is being used, i.e., variable of x and y are considered with x_1 and x_2 respectively. In this method at every point of implementation, symbolic value of program variable and path constraint are presented as a rational formula on the symbolic values of the program variables. For access to that point, the path constraints must be "true". In addition, the path constraints are determined by the logical expressions used in the branches, which are updated with each branch. Any combination of real inputs, for which the value of the path constraint is "true", it could be considered as a program input that guarantees the execution of the desired path. This method must use constraint solvers to find the actual values in order to produce the test case. These approaches can determine infeasible paths simply. In these methods, constraints solvers have been used to find the actual values in order to produce the test case. Therefore, the efficiency of the method is strongly dependent upon the efficiency of solver and the calculation of host hardware. Moreover, in case of non-linear branch conditions, static methods have significant overhead cost.

Dynamic methods involve in testing the software for the input values and analyze the output values according to the generated input. In fact, dynamic methods generate input values for program under test. Dynamic methods comprise random testing, local search approach [8], goal-oriented approach [5], chaining approach [9] and evolutionary approach [9–13]. In these methods, the software is tested by inserting inputs and measuring the number of target paths covered by the software. Moreover, due to predefined of input variables determined during the execution of the program, the production of dynamic test data can prevent those problems encountered by static methods.

Hybrid methods combine the advantages of static methods (like reducing domain of problem) with the benefits that can be obtained from the dynamic methods (such as reducing the costs), combination methods have been developed [17].

All of method evaluations are based on different criteria. There are different test criteria, such as instruction coverage, branch coverage and path coverage.

Instructions Coverage: In this case, it is necessary to select input data from the problem space that all instructions are executed at least once.

Branch Coverage: The input data is selected from the problem space that all the branches are executed at least once [3].

mmeybodi@aut.ac.ir

Path Coverage: The input data is selected from the problem domain that all the paths are traversed at least once.

This paper addresses path coverage. In particular, consider the most difficult paths. It used the hybrid method that the symbolic execution as static method and evolutionary algorithm as dynamic method selected to generate test data generation.

In this paper, one of the most recent works in the field of static and dynamic methods for test data generation has been improved. In [14], by combining the previous fitness functions and improving them, they developed a new fitness function for the GA. In our approach, by using the proposed silent function, as well as changing in the main architecture of the GA, a new approach is developed. The proposed method has been experimented on the 7 standard benchmarks introduced in [21]. The results and performance demonstrated a significant improvement in the efficiency and effectiveness of the software testing.

The remainder of this paper is organized as follows: The second section and the third section introduce background and related work in this area, respectively. GA and our approach in detail are presented in the fourth section. In the five section, the proposed method is applied to standard benchmarks and provided the illustrative experiments that compared with recent papers and the last section gives the conclusion and future work to the paper.

2 Background

Most of fitness functions in software testing research area are based on approach level [15] and branch distance [16] which are two approaches to calculate generated test cases fitness functions. Approach level was proposed [15] and calculate test cases fitness function by enumerating remained branches to execute to gain the target branch. Branch distance factor is the test case's distance from satisfying a branch's condition. In other word, a number must be added or subtracted from the test case to satisfy the condition. Consequently, this two-fitness factor combine together to improve fitness functions accuracy which calculate by following equation:

$$f_{AL}(i) = level(b) + \eta(i, b)$$

In above equation $level(b)$ is approach level and $\eta(i, b)$ is the branch distance.

Discussed approaches did not consider executed branches, therefore Symbolic Enhanced Fitness Function was proposed by Harmen et al. at [17]. They add a simple static analysis. i.e., symbolic executor to evolutionary algorithms for software testing. It calculates the cost of that a test case can satisfy all branch conditions with a normalized branch distance. By mean that this approach attends all executed and non-executed branches. This approaches equation is calculating according to the following equation:

$$f_{SE}(i) = \sum\nolimits_{b \epsilon P} \eta(i, b)$$

In [14] by portion of Symbolic Enhanced Fitness Function, proposed a factor for determining branches hardness level and calculated test cases finesses according to the

mmeybodi@aut.ac.ir

branches harnesses. They formulated the hardness, considering two main factors, first one, number of variables in the branch condition(α(c)) which extracted by Symbolic analyzer and second one is the branch conditions tightness(β(c)) Which is ratio of number of solutions in the problem's domain to the size of domain. It also used a reinforcement coefficient to tunes effect of these two discussed factors in calculation of branches hardness. their hardness factor is calculated as follow:

$$\mathrm{DC(c)} = B^2 \times \alpha(\mathrm{c}) + \mathrm{B} \times \beta(\mathrm{c}) + 1$$

This hardness is as a punishment to test cases who cannot satisfy the branch. And its related fitness function calculation is as the following equation:

$$f_{DC}(i, C) = \sum_{c \epsilon C} DC(c) \times \eta(i, b)$$

For example consider $i_1 = (10, -30, 60)$, $i_2 = (30, -20, -20)$ as two test cases and Fig. 1 as our source code.

```
1  function(int x, int y, int z){
2        if( y==z )
3           if( y>0 )
4              if( x==10 )
5                 ...//Target
6 }
```

Fig. 1. Example source code.

There are three branches in this source code in lines: 2, 3 and 4. This program branches hardness's has been calculated as:

DC("y==z") = $10^2 \times 0.5 + 10 \times 0.995 + 1 = 60.95$

DC("y>0") = $10^2 \times 1 + 10 \times 0.5 + 1 = 106$

DC("x=10") = $10^2 \times 1 + 10 \times 0.995 + 1 = 110.95$

Therefore i_1 and i_2 finesses would be:

$$f_{DC}(i_1, C) = 60.95 \times \frac{90}{91} + 106 \times \frac{31}{32} + 110.95 \times \frac{0}{1} = 162.9677$$

$$f_{DC}(i_2, C) = 60.95 \times \frac{0}{1} + 106 \times \frac{21}{22} + 110.95 \times \frac{20}{21} = 206.8485$$

According to their fitness values i_1 had been preferred than i_2.

mmeybodi@aut.ac.ir

3 Related Work

In this section, we review the most important methods that centered around different meta-heuristic algorithms.

In [14] benefited from both static and dynamic approaches advantages. it extracts some information from path conditions using static analyzing. the information had been used for defining more exact population instead of random initialization of the first population for GA.

After 2014 most of researchers concentrate on guiding GA to faster converge which that leaded to decrease in calculation costs. Accordingly, to that designing an appropriate fitness function considered by researchers. In [14] proved that branches have no equivalent values according to their hardness. It means that satisfying a harder branch is more valuable, therefore a test case who satisfies harder branches is more valuable. So they had been defining hardness factor to determining each branch harnesses, which has been used in fitness function equation [18].

In [13] an approach to improve GA efficiency proposed. They defined their exclusive branch distance and fitness function. In addition [1] reinforced GA by considering a preprocessing step before performing the algorithm. They extracted hard path conditions and used them to made a kind of adjustment for GA which tunes individuals for faster converging. [19] combined static and dynamic approaches to generating test cases, they developed their static analyzer (JDBC) to extract path conditions, and used a search problem converter that converts extracted path conditions to optimization problems and finally they use GA to solve these optimization problems. In [20] a branch hardness factor defined using probability of visits, hence branches with fewer Expected number of visits are harder than other.

4 Proposed Approach

This section depicts details of our proposed approach, to generate test cases for path coverage using augmented GA. By using the proposed silent function in [14], and changing in the main architecture of the GA, a new approach is developed in the field of automatic test data generation.

Generally speaking, evolutionary algorithms search for a general optimal point in the solution space, and usually cannot search locally around specific responses [22]. They could be trapped in an optimal point. In addition, sample space of software testing problem is very extensive. Therefore, this problem would be obvious. Have the feature of evolutionary algorithms (general search) is combined with a local search algorithm, the results will be improved. In other words, the evolutionary algorithm first finds good answers. Then, this area could be accurately searched by a local search algorithm to find the optimal point. Details of our approach is described below.

Genetic algorithm is a search heuristic that is inspired from Charles Darwin's theory of natural evolution. This algorithm models the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring for the next generation. The process of natural selection starts with the selection of fittest individuals from a population. They generate individuals that almost keep the characteristics of their parents and will be added to the next generation. If parents are

mmeybodi@aut.ac.ir

fitter, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found. GA has a wide application in optimization problems [23].

Based on the Fig. 2(a), the GA architecture consists of six phases:

1. Initial population to start the algorithm.
2. Population Fitness functions evaluation and assign a fitness number to each individual.
3. Selection: select a pair of individuals as parent to make offspring.
4. Crossover: is evolution operator which exchange parents' bits with together to generates better individuals.
5. Mutation: mutate some bits to avoiding trapping in local optimums.
6. Replacement: replace new generated population with old one.

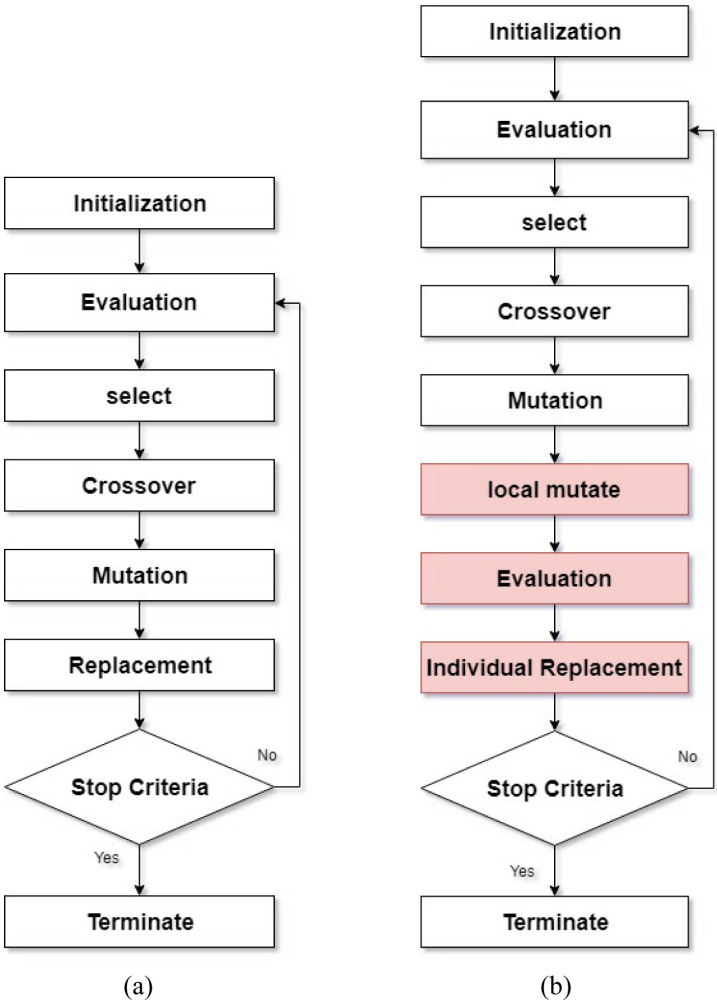


Fig. 2. (a), (b) show the architecture of traditional GA and augmented GA, respectively.

mmeybodi@aut.ac.ir

In our proposed architecture showed in Fig. 2(b), in addition to the above steps, two new steps are added in which selection and mutation operators are re-evaluated and appropriate operators selected. The basis of this algorithm is inspired by the hill climbing algorithm, therefore, it could be defined as a local search algorithm.

Local Search. The algorithm, among the neighbors of each individual, probs the fittest point. To calculate neighborhood of Individual k, D-dimensional space is considered. The neighbors of Individual k with position vector $IND_k = (x_{k1}, x_{k2}, \ldots, x_{kd})$ have a new position vector of $IND'_k = (x'_{k1}, x'_{k2}, \ldots, x'_{kd})$ where $x'_{k1} = x_{k1} + p$, $-500 < p < +500$ and $x'_{k1} \neq x_{k1}$ that p based on a gaussian distribution is selected.

The rule for local transfer of Individual location would be depicted as follows: Individual k transfers from x_k to a new location x'_k if the fitness of x'_k is better than that of x_k (i.e., fitness(x'_k) > fitness(x_k)), and x'_k has the best fitness value among x'_k neighbors. Otherwise, the Individual k must stay at its current location (i.e., x_k).

5 Experimental Results

We implemented [14] as a base and improved this approach. Our proposed algorithm ran on 7 standard benchmarks. it is Noteworthy that we had 30 runs on each benchmark and all of presented data is averaged out 30 times of run. We compared our approach with three others according to 2 factors, coverage percentage of targets in the benchmarks and Average Time Cost (ATC) of running of each benchmark which has been calculated using this formula:

$$\mathrm{ATC} = \frac{1}{|S|} \sum_{i \epsilon S} TC_i$$

In the above equation S is the set of successful runs of the algorithm. And TC is the time cost of each run individually. ATC determines the fair time cost for the algorithm (Table 1).

Table 1. A comparison between this paper approach and others [20].

Benchmarks	Fitness function approaches							
	Proposed approach		Sakti [14]		Symbolic EXE		Approach level	
	Coverage	ATC(s)	Coverage	ATC(s)	Coverage	ATC(s)	Coverage	ATC(s)
Gammaq	100%	0	100%	0	66%	0.370	59%	0.309
Expint	100%	2.133	75%	2.158	31%	2.180	1%	1.495
Ei	100%	0.133	75%	0.597	77%	0.947	77%	0.685
Bessj	100%	0.541	60%	2.103	31%	2.240	6%	1.059
Bessi	100%	0.539	85.5%	2.001	51%	1.978	11%	1.406
Plgndr	100%	0	–	–	0%	–	0%	–
Betai	100%	0	100%	1.259	70%	1.115	13%	0.938

mmeybodi@aut.ac.ir

Our results clearly prove this approach's superiority than former approaches.

In the following diagram we can see the speed of convergence of proposed approach against other former approaches. Figure 3 shows the percentage of coverage in the number of generations produced in five different approaches. As we can see, number of generation that our proposed approach needed to completely cover all targets is far less than other approaches. While other approaches in the number of generations more than our attitude have reached 80% coverage, none of them have been able to fully cover 54 goals.

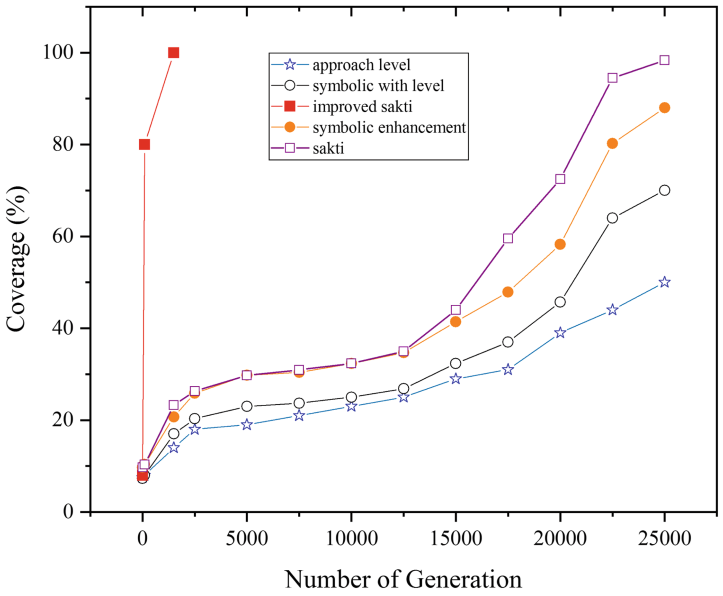


Fig. 3. Show coverage rate of different approaches vs our approach.

Tuning parameters of these papers are according to the following (Table 2):

Table 2. Implementation details

Bounds	[−1000, 1000]
Population	100
Mutation rate	0.5
Number of comparisons in local search for each individual	2 per each gene

mmeybodi@aut.ac.ir

6 Conclusion and Future Work

In this paper, we proposed a search-based test data generation approach to cover Paths coverage of the program under test. By using the proposed silent function in [14], as well as improving in the main architecture of the GA. The experimental results of some programs under test demonstrated that augmented GA generated test data can cover all feasible paths having path conditions which cannot be covered by test data generated from regular GA. The main reason for this superiority is due to the local search.

Since these issues are inherently different from optimization issues, and in most cases the level of response space is discrete, the combination of search optimization algorithms such as linear programming with this algorithm can be very useful. There have been some studies performed in this area that, definitely, should be used as a function of this combination. (i.e., in the initialization step, some parts of the answer can be obtained with precise methods).

References

1. Dinh, N.T., Vo, H.D., Vu, T.D., Nguyen, V.H.: Generation of test data using genetic algorithm and constraint solver. In: Asian Conference on Intelligent Information and Database Systems, pp. 499–513. Springer, Cham (2017)
2. Myers, G.J.: The Art of Software Testing (1979)
3. Xibo, W., Na, S.: Automatic test data generation for path testing using genetic algorithms. In: Third International Conference on Measuring Technology and Mechatronics Automation, pp. 596–599 (2011)
4. James, C.K.: A new approach to program testing. In: Proceedings of the International Conference on Reliable Software. ACM, Los Angeles (1975)
5. Chen, T.Y., Tse, T.H., Zhou, Z.: Semiproving: an integrated method based on global symbolic evaluation and metamorphic testing. In: International Symposium on Software Testing and Analysis. ACM, Roma (2002)
6. Sy, N.T., Deville, Y.: Consistency techniques for interprocedural test data generation. ACM SIGSOFT Softw. Eng. Notes **28**, 108–117 (2003)
7. Michael, C.C., McGraw, G., Schatz, M.: Generating software test data by evolution. IEEE Trans. Softw. Eng. **27**, 1085–1110 (2001)
8. Korel, B.: Automated software test data generation. IEEE Trans. Softw. Eng. **16**, 870–879 (1990)
9. Korel, B.: Automated test data generation for programs with procedures. In: Proceedings of the 1996 ACM SIGSOFT International Symposium on Software Testing and Analysis. ACM, San Diego (1996)
10. Xanthakis, S., Ellis, C., Skourlas, C., Le Gall, A., Katsikas, S., Karapoulios, K.: Application of genetic algorithms to software testing. In: Proceedings of 5th International Conference on Software Engineering and Its Applications, Toulouse, France, pp. 625–636 (1992)
11. Wegener, J., Baresel, A., Sthamer, H.: Evolutionary test environment for automatic structural testing. Inf. Softw. Technol. **43**, 841–854 (2001)
12. Wegener, J., Buhr, K., Pohlheim, H.: Automatic test data generation for structural testing of embedded software systems by evolutionary testing. In: Proceedings of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann Publishers Inc. (2002)

mmeybodi@aut.ac.ir

13. Thi, D.N., Hieu, V.D., Ha, N.V.: A technique for generating test data using genetic algorithms. In: International Conference on Advanced Computing and Applications. IEEE Press, Can Tho (2016)
14. Sakti, A., Guéhéneuc, Y.G., Pesant, G.: Constraint-based fitness function for search-based software testing. In: International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Springer, Heidelberg (2013)
15. Tracey, N., Clark, J.A., Mander, K., McDermid, J.A.: An automated framework for structural test-data generation. In: ASE, pp. 285–288 (1998)
16. Arcuri, A.: It does matter how you normalise the branch distance in search based software testing. In: ICST, pp. 205–214. IEEE Computer Society (2010)
17. Baars, A.I., Harman, M., Hassoun, Y., Lakhotia, K., McMinn, P., Tonella, P., Vos, T.E.J.: Symbolic search-based testing. In: Alexander, P., Pasareanu, C.S., Hosking, J.G. (eds.) ASE, pp. 53–62. IEEE (2011)
18. Sakti, A.: Automatic Test Data Generation Using Constraint Programming and Search Based Software Engineering Techniques. École Polytechnique de Montréal (2014)
19. Braione, P., et al.: Combining symbolic execution and searchbased testing for programs with complex heap inputs. In: Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis. ACM (2017)
20. Xu, X., Zhu, Z., Jiao, L.: An adaptive fitness function based on branch hardness for search based testing. In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM (2017)
21. http://www.crt.umontreal.ca/∼quosseca/fichiers/23benchsCPAOR13.zip
22. Yao, X.: Evolving artificial neural networks. Proc. IEEE **87**(9), 1423–1447 (1999)
23. https://towardsdatascience.com/introduction-to-geneticalgorithms-including-example-code-e396e98d8bf3

mmeybodi@aut.ac.ir

Building and Exploiting Lexical Databases for Morphological Parsing

Petra Steiner[1(✉)] and Reinhard Rapp[2]

[1] Institute of German Linguistics, Friedrich-Schiller-Universität Jena, Fürstengraben 30, 07743 Jena, Germany
petra.steiner@uni-jena.de
[2] Hochschule Magdeburg-Stendal, Breitscheidstraße 2, 39114 Magdeburg, Germany
reinhard.rapp@hs-magdeburg.de

Abstract. This paper deals with the use of a new German morphological database for parsing complex German words. While there are ample tools for flat word segmentation, this is the first hybrid approach towards deep-level parsing of German words. We combine the output of the two morphological analyzers for German, Morphy and SMOR, with a morphological tree database. This database was created by exploiting and merging two pre-existing linguistic databases. We describe the state of the art and the essential characteristics of both databases and their revisions.

We test our approach on an inflight magazine of Lufthansa and find that the coverage for the lemma types reaches up to 90%. The overall coverage of the lemmas in text reaches 98.8%.

Keywords: Lexical databases · German · Morphology · Compounding · Derivation

1 Introduction

German is a language with complex processes of word formation, of which the most common are compounding and derivation. Segmentation and analysis of the resulting word forms are challenging as spelling conventions do not permit spaces as indicators for boundaries of constituents as in (1).

(1) Felsformation ‘rock formation’

For long orthographical word forms, many combinatorially possible analyses exist, though usually only one of them has a conventionalized meaning (see Fig. 1). There are many ambiguous boundaries. For *Felsformation* ‘rock formation’, word segmentation tools can yield the wrong split containing the more frequent word tokens *Fels* ‘rock’, *Format* ‘format’, and *Ion* ‘ion’.

Often homonyms of free and bound morphemes pose problems. Figure 2 shows the deep analyses for (1) where the string *ion* is a bound morph of the loan word *Formation* and not interpretable as the free morph *Ion* ‘ion’.

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 258–273, 2020.
https://doi.org/10.1007/978-3-030-37309-2_21

mmeybodi@aut.ac.ir

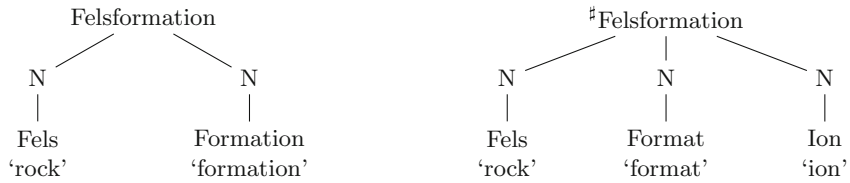


Fig. 1. Ambiguous analysis of *Felsformation* ‘rock formation’

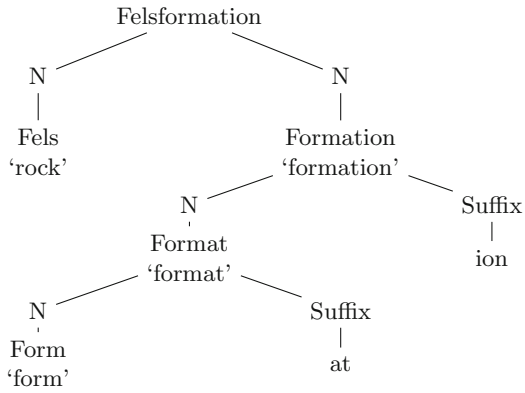


Fig. 2. Deep analysis of *Felsformation* ‘rock formation’ according to the CELEX database

Combinatorially, the number of possible analyses for each word segmentation is identical to the number of compositions of the number of the smallest components. This number has to be multiplied by the number of homonyms for the segmented forms. Therefore, automatic segmentation with more than ten possible analyses for one word are not a rare case.

However, finding the correct segmentations and morphological structures is essential for terminologies and translation (memory) tools, information retrieval, and as input for textual analyses. Moreover, frequencies of morphs and morphemes are required for testing quantitative hypotheses about morphological tendencies and laws.

In this paper, we will be using a hybrid approach for finding the correct splits of words. In Sect. 2, we provide a concise overview of previous work in word segmentation and word parsing for German. We introduce three linguistic tools in Sect. 3. These are the morphological tool *SMOR*, its add-on module *Moremorph*, and the morphological tool *Morphy*. Section 4 introduces our morphological database which was built on the basis of the linguistic databases CELEX and GermaNet. Section 5 describes our procedures for the morphological analyses. In Sect. 6, we test three variants of simple and hybrid approaches. Finally, we discuss our results and give an outlook for future developments.

mmeybodi@aut.ac.ir

2 Related Work

The first morphological segmentation tools for German were developed in the nineties and most of them are based on finite state machines. GERTWOL [9], MORPH [11], Morphy [17,18,20], and later SMOR [26] and TAGH [7] can generate an abundance of analyses for relatively simple words.

There are some ways to solve this ambiguity problem: One is using ranking scores, such as the geometric mean, for the different morphological analyses [3,14] and then choosing the segmentation with the highest ranking. Another consists in exploiting the sequence of letters, e.g. by pattern matching with tokens [12, p. 422], [31], or lemmas [32]. Candidates of compound splits can also be obtained by string comparisons with corpus data [4,31]. [31] combine this method with a ranking score based on frequencies of the strings of hypothetical components within tokens in a large corpus. However, this method fails for cases of ambiguity with one word string completely embedded into the other one (e.g. *Saal* 'hall' vs. *Aal* 'eel'). Combining normalization with ranking by the geometric mean is another method [35]. Furthermore, Conditional Random Fields modeling can be applied for letter sequences [21].

Recent approaches exploit semantic information for the ranking. [23] combine a compound splitter and look-ups of similar terms inside a distributional thesaurus generated from a large corpus. [34] use the cosine as a measure for semantic similarity between compounds and their hypothetical constituents. They compute the geometric means and other scores for each produced split. These scores are then multiplied by the similarity scores. Thus, a re-ranking is produced which shows a slight improvement.

Most tools for word analyses of German word forms provide flat sequences of morphs but no hierarchical parses which could give important information for word sense disambiguation. Restricting their approach to adjectives, [33] are using a probabilistic context free grammar for full morphological parsing. [29] developed a method for building parts of morphological structures. They reduced the set of all possible low-level combinations by ranking morphological splits with the geometric mean.

[34] discuss left-branching compounds consisting of three lexemes such as *Arbeitsplatzmangel* (Arbeit|Platz|Mangel) '(work|place|lack) job scarcity'. Their distributional semantic modelling often fails to find the correct binary split if the head (here *Mangel* 'lack') is too ambiguous to correlate strongly with the first part (here *Arbeitsplatz* 'employment') though in general, using the semantic context is a sensitive disambiguation method. [35] use normalization methods. Their segmentation tool can be used recursively by re-analyzing the results of splits.

All these approaches build strongly upon corpus data but none of them uses lexical data. Only [12] enrich the output of morphological segmentation with information from the annotated compounds of GermaNet. This can in a further step yield hierarchical structures but presupposes that the entries for the components exist inside the database. In Sect. 4.3, we come back to this strategy and will exploit the GermaNet database, and CELEX as another lexical resource.

mmeybodi@aut.ac.ir

3 Two Morphological Splitters

3.1 SMOR: A Morphological Tool for German

SMOR is a widely used morphological segmentation tool (e.g. [3,12,29,34]). It is based on two-level morphology [15] and implemented as a set of finite-state transducers. For German, a large set of lexicons is available. These lexicons contain information about inflection, parts of speech, and classes of word formation, e.g. abbreviations and truncations. The tag set used is compatible with the STTS (Stuttgart Tübingen tag set [24]).

SMOR produces different levels of granularity and different representation formats with different transducers and options. Example (2) shows a typical output of fine-grained analyses:

(2) Fels<NN>Formation<+NN><Fem><Acc><Sg>
Fels<NN>Formation<+NN><Fem><Dat><Sg>
Fels<NN>Formation<+NN><Fem><Gen><Sg>
Fels<NN>Formation<+NN><Fem><Nom><Sg>
Fels<NN>format<V>ion<SUFF><+NN><Fem><Acc><Sg>
Fels<NN>format<V>ion<SUFF><+NN><Fem><Dat><Sg>
Fels<NN>format<V>ion<SUFF><+NN><Fem><Gen><Sg>
Fels<NN>format<V>ion<SUFF><+NN><Fem><Nom><Sg>

Here, the word form *Felsformation* 'rock formation' is analyzed in eight different ways, without the erraneous interpretation of *ion* as a noun. The categories show parts of speech (<NN>, <V>) of free morphs, the position of bound morphemes (<SUFF>), and the case and number of the analyzed word. Please note that *format* is interpreted as a verbal stem here.

An analysis with a minimal number of constituents can also be produced. This format is a much-used standard in morphological analyses with SMOR. See (3) for an output of the immediate constituents.

(3) Fels<NN>Formation<+NN><Fem><Acc><Sg>
Fels<NN>Formation<+NN><Fem><Dat><Sg>
Fels<NN>Formation<+NN><Fem><Gen><Sg>
Fels<NN>Formation<+NN><Fem><Nom><Sg>

3.2 Moremorph: The Add-On for SMOR

Moremorph aims at improving and adjusting the output of SMOR for the following:

Lexical Bottlenecks. In general, not every word form can be analyzed by SMOR, as the lexicons are limited. Of course, not every name of small Iranian or German villages can be expected to be recognized. However, for some restricted domains this might be required. To improve the recall, we extended

mmeybodi@aut.ac.ir

the original lexicons and the transition rules. The original version of the names lexicon comprised 14,998 entries, the final extended version 16,718 entries.

The original general lexicon was obtained from Helmut Schmid. During the period of the project, he also removed and added lemmas. The last version which was obtained comprised 41,941 entries. Some information of the lexicons is redundant or can prevent expected analyses, especially if complete compounds do exist as lexical entries. Therefore, size does not necessarily imply quality. If lemmas are flagged as only initial in compounds or not as constituents at all, this can yield or prevent mistakes. Therefore, refining such information was also essential. During the project, the lexicon was constantly extended and cleaned and its entries were revised. The final version used for the current work comprises 42,205 entries.

Many changes of the rule sets were made in cooperation with Helmut Schmid according to our suggestions. For example, we changed the sets of characters or added adverbs as possible tag class for numbers. Other changes include the derivation of adjectives from names of locations. Often more than one transducer had to be changed.

Special Characters Inside Words. In SMOR, sequences of unknown constituents between hyphens are generally analyzed as truncated parts, see (4) for *Lut-Felsformation*[1]. This leads to inconsistent analyses for orthographical variants with and without hyphenations.

(4) Lut-<TRUNC>Fels<NN>Formation<+NN><Fem><Nom><Sg>
Lut-<TRUNC>Fels<NN>Formation<+NN><Fem><Gen><Sg>
Lut-<TRUNC>Fels<NN>Formation<+NN><Fem><Dat><Sg>
Lut-<TRUNC>Fels<NN>Formation<+NN><Fem><Acc><Sg>

A similar problem emerges for forms with special characters such as *Köln/Bonn* 'Cologne/Bonn' which yields no result, whereas (5) can be analyzed at least. This results in inconsistent analyses for orthographical variants such as *Flughafen Köln-Bonn* 'Airport Cologne-Bonn' vs. *Flughafen Köln/Bonn* 'Airport Cologne/Bonn'.

(5) Köln-<TRUNC>Bonn<+NPROP><Neut><Nom><Sg>
Köln-<TRUNC>Bonn<+NPROP><Neut><Dat><Sg>
Köln-<TRUNC>Bonn<+NPROP><Neut><Acc><Sg>

Other examples not covered by SMOR are listed in (6).

(6) a. "Team Lufthansa"-Partner
b. "buy & fly"-Angebote "'buy & fly" offers'

Hyphens and forms with similar functions are being treated by three methods: a. We generate templates without these characters, send them to the analysis and re-insert the characters. b. Parts which were analysed with the tag TRUNC are

[1] Dasht-e-Lut: salt desert in Iran.

mmeybodi@aut.ac.ir

reanalyzed. c. If an analysis with the template method does not yield a result, the re-analysis will be invoked for strings between hyphens and functionally similar characters. All such characters will be tagged with the tag HYPHEN as in (7):

(7) Köln/Bonn Köln / Bonn NPROP HYPHEN NPROP <NPROP>

A special case are words which are beginning or ending with the characters -or/as in (8). In these cases, these characters are simply stripped and not re-inserted. If there is a filler letter such as *s* in (8-a), this is stripped too. Some other tags are also removed from the SMOR output, e.g. the meta-tag ABBR for abbreviations.

(8) a. *Abfertigungs-* $\Longrightarrow$ *Abfertigung* 'clearance'
b. *und/* $\Longrightarrow$ *und* 'and'

Unanalyzed Interfixes. Furthermore, the SMOR output does not indicate if there are filler letters (or interfixes) inside a word.[2] In example (9) for *wirkungsvoll* 'effect|*filler_letter*|full, effective', the interfix between *Wirkung* and *voll* has been deleted by SMOR.

(9) Wirkung<NN>voll<+ADJ><Pos><Pred>

However, the information exists inherently in the intermediate SMOR output. Therefore, they can be marked as such as in *wirkungsvoll* 'effect|*filler_letter*|full, effective' in (10).

(10) wirkungsvoll W:wirkung s voll NN FL ADJ <ADJ>

3.3 Morphy

As described in [17–20], Morphy is a freely available tool for German morphological analysis, generation, part-of-speech tagging and context sensitive lemmatization. The morphological analysis is based on the Duden grammar and provides wide coverage with a lexicon of 50,500 stems which correspond to about 324,000 full forms. Requiring less than 2 Megabytes of storage, Morphy's lexicon is very compact as it only stores the base form of each word together with its inflectional class. New words can be easily added to the lexicon via a user-friendly input system.

In its generation mode, starting from the root form of a word, Morphy looks up the word's inflectional class as stored in the lexicon and then generates all inflected forms. In contrast, Morphy's analysis mode is used for analyzing text. In this mode, for each word form found in a text, Morphy determines its root, part of speech, and – as appropriate – its gender, case, number, person, tense, and comparative degree. If a context analysis is desired, tagging mode is available

[2] By some approaches, such interfixes are considered as a special kind of morphemes and called *Fugenmorpheme* 'linking elements'. We like to avoid such classifications and use the labels *filler letters* or *interfix*.

mmeybodi@aut.ac.ir

where Morphy selects the supposedly best morphological description of a word given its neighboring words [22]. However, as Morphy's standard feature system with a large tag set of 456 tags is sophisticated, an accuracy of only about 85% can be expected in this mode. But in cases where such sophistication is not required, it is possible to switch to a smaller tag set where the number of features under consideration is reduced. This tag set of 51 tags is comparable in size to the standard tag sets used by part-of-speech taggers for other languages, and it achieves a similar accuracy of about 96%.

For the purpose of this paper, the most important feature of Morphy is its capability of segmenting complex words. In contrast to SMOR, Morphy also takes interfixes into account. The underlying algorithm is based on a longest match procedure which works from right to left. That is, the longest noun base form or full-form as found in the lexicon is matched to the right side of any unknown word form as occurring in a text, thereby presupposing that this unknown word form might be a compound noun. If the matching is successful, this procedure can be repeated several times, until no more matching is achieved.

In this way, the split in Fig. 1 would be chosen correctly. It should be noted, however, that occasionally the preference for long matches can lead to incorrect results. An example is *Arbeitsamt* 'job center', which by this procedure will be incorrectly interpreted as *Arbeit-Samt* 'work velvet', instead of *Arbeit* 'work' - *filler_letter* - *Amt* 'office'. (11) shows a typical output of Morphy with the lemmatized form *Felsformation* and its morphosyntactic information.

(11) Felsformation
Felsformation SUB NOM PLU FEM KMP Fels/Formation
Felsformation SUB GEN PLU FEM KMP Fels/Formation
Felsformation SUB DAT PLU FEM KMP Fels/Formation
Felsformation SUB AKK PLU FEM KMP Fels/Formation

4 Two Lexical Databases with Morphological Information

While syntactic treebanks for German have existed for many years, to our knowledge there is no such kind of data for morphology, besides some mostly internally used gold standards (e.g. the test set of the 2009 workshop on statistical machine translation[3] which was used by [3]). While it would be an honorable task to augment such existing flat structures or produce new morphological data, this is very cumbersome and time-consuming. Therefore, we preferred to look for recyclable resources from which complex syntactic structures could be derived automatically. We found two resources: a. the CELEX database for German morphology, b. the compound analyses from the GermaNet database.

Before we could process these databases, the morphological information of both sources had to be changed according to our needs. For both modified datasets, the derivation of complex structures was performed recursively. Because

[3] http://www.statmt.org/wmt09/translation-task.html.

mmeybodi@aut.ac.ir

of the structure of the data and certain kinds of errors it contains, we set restrictions and used heuristics for inferring the data format we need. Finally, we combined the GermaNet analyses with the analyses we obtained from CELEX. In the following subsections, we describe the original data, their modifications and their merging.

4.1 CELEX

The CELEX database [1] is a lexical database for Dutch, English, and German [2]. In addition to information on orthographic, phonological and syntactic features, it also contains ample information on word-formation, especially manually annotated multi-tiered word structures. Though old, it still is one of the standard lexical resources for German. The linguistic information is combined with frequency information based on corpora [8, p.102ff.]. The morphological part comprises flat and deep-structure morphological analyses of German, from which we will derive treebanks for our further applications.[4]

As the database was developed in the early nineties, it has some drawbacks: Both encoding and spelling are outdated. About one fifth of over 50,000 datasets contain umlauts such as the non-ASCII letters *ä* or *ö*, and signs such as *ß*. These letters are represented by ASCII substitutes such as *ae* for *ä* or *ss* for *ß*.

Another problem is the use of an outdated spelling convention which makes the lexicon partially incompatible with texts written after 1996 when spelling reforms were implemented in Austria, Germany and Switzerland. For instance, the modern spelling of the originally CELEX entry *Abschluß* 'conclusion' is *Abschluss*.

As the database was created according to the standardized spelling conventions of its time, there are only a few spelling mistakes which call for corrections. [27] describes how the data was transformed to a modern standard.[5]

(12) presents a typical entry of the refurbished CELEX database for the lexeme *Abdichtung* '*prefix*, dense, *suffix* = sealing'.

(12) 87\Abdichtung\3\C\1\Y\Y\abdicht+ung\Vx\N\N\N\
(((ab)[V|.V],(dicht)[V])[V],(ung)[N|V.])[N]\N\N\N\N\S3/P3\N

Here the tree structure can be directly recognized within the parenthetical structure. However, this is not always the case. For instance, in (13) *Abbröckelung* 'crumbling', the complete derivation comprises a derived verb *bröckeln* 'to crumble' of the noun *Brocken* 'crumb'. This is not evident from the entry.

Some derivations in the German CELEX database provide diachronic information which is correct but often undesirable for many applications, for example in *Abdrift* 'leeway' (14) which is diachronically derived from *treiben* 'to float'.

(13) 63\Abbröckelung\0\C\1\Y\Y\Y\abbröckel+ung\Vx\N\N\N\
(((ab)[V—.V],(((Brocken)[N])[V],(el)[V—V.])[V])[V],(ung)[N—V.])[N]
[...]

[4] For an exhaustive description of the German part of the database see [8].
[5] See https://github.com/petrasteiner/morphology for the script.

mmeybodi@aut.ac.ir

(14) 97\Abdrift\0\C\1\Y\Y\Y\ab+drift\xV\N\N\N\
((ab)[N—.V],((treib)[V])[V])[N]\Y\N\N\N\S3/P3\N

(15) 605\\Abschlussprüfung\\C\1\Y\Y\Y\Abschluss+Prüfung\\NN\N\N\N\
((((ab)[V|.V],(schließ)[V])[V])[N], ((prüf)[V],(ung)[N|V.])[N] [...]

(16) 207\Abgangszeugnis\4\C\1\Y\Y\Y\Abgang+s+Zeugnis\NxN\N\N\N\
((((ab)[V—.V],(geh)[V])[V])[N],(s)[N—N.N],((zeug)[V],(nis)[N—V.])[N])[N]
[...]

On the other hand, some derivations such as the ablaut change between *Schluss* 'end' and *schließen* 'to finish' in *Abschluss* (15), or the one between *gehen* 'to go' and *Gang* 'gait,path,aisle' in *Abgangszeugnis* 'leaving certificate' (16) in Fig. 3 could be of interest.

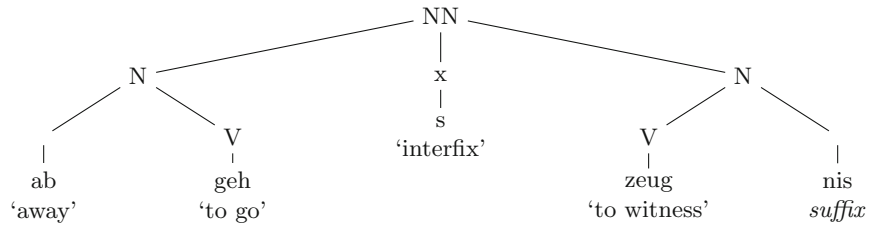


Fig. 3. Morphological analysis of *Abgangszeugnis* 'leaving certificate' as in the refurbished CELEX database

4.2 GermaNet

GermaNet [10] is a lexical-semantic database which in principle is compatible with Princeton WordNet [6]. In addition, it comprises information which is specific for the German language such as noun inflection or particle verbs. [12] augmented the GermaNet database with information on compound splits. However, this is restricted to nouns and does not provide interfixes or deep-level structures. The data was revised since then and we are using the version 11 which was updated in February 2017 for the last time.[6] (17) presents the entry for *Abgangszeugnis* 'leaving certificate'. As can be seen, the interfix *s* is missing in the analysis.

(17) <lexUnit id="l41391" sense="1" source="core" namedEntity="no"
artificial="no" styleMarking="no">
<orthForm>Abgangszeugnis</orthForm>

[6] See http://www.sfs.uni-tuebingen.de/GermaNet/compounds.shtml#Download for a description.

mmeybodi@aut.ac.ir

<compound> <modifier category="Nomen">Abgang</modifier>
<head>Zeugnis</head> </compound> </lexUnit>

4.3 Building and Merging Morphological Trees of the Databases

We extract and preprocess all relevant information from both databases, such as all immediate constituents and their categories. For each entry of the respective morphological database, the procedure starts from the list of its immediate constituents and recursively collects all information.

For coping with dissimilar word stems in diachronic derivations in CELEX, we calculate the Levenshtein distance (LD) for the strings s_1, s_2 of the smaller length of the two compared constituents ($min(l_1, l_2)$), and then compare their quotient *dis* to a threshold t as in Eq. 1.[7] We also added a small list of exceptions.

$$dis = \frac{LD(s_1, s_2)}{min(l_1, l_2)} \leq t \tag{1}$$

For GermaNet (GN), we remove proper names and foreign word expressions, furthermore, we add interfixes by heuristics.

We generated morphological analyses of both databases (CELEX trees and GN trees). The data from GermaNet is restricted to compound nouns which can be complex and special terms. On the other hand, CELEX trees comprise not only compounds but also deep-level analyses of derivatives and conversions which cover most lexemes of German basic vocabulary. Therefore, we decided to combine both sets, by starting with a recursive look-up in GermaNet which is augmented by CELEX trees as soon as the look-up stops and vice versa. The algorithms can be found in [28]. Different depths of the structures from flat to very fine-grained can be produced by setting respective flags. Finally, both complex sets were unified. In a final step, we added the 11,100 simplex words of CELEX for the recognition of non-analyzable words such as *Fels* 'rock'. (18) shows the morphological structures with categorial information of *Abschlussprüfung*, *Abdrift*, and *Abgangszeugnis* for a Levenshtein threshold of 0.75.

(18) a. Abschlussprüfung (*Abschluss_N*
(*abschließen_V* ab_x| schließen_V))|
(*Prüfung_N* prüfen_V| ung_x)
b. Abdrift ab_x| (driften_V)
c. Abgangszeugnis (*Abgang_N* (*abgehen_V* ab_x| gehen_V))| s_x|
(*Zeugnis_N* (zeugen_V| nis_x)

Table 1 shows the number of entries for the databases of the morphological trees. Double entries were removed.

[7] [30] provides an example for this heuristics.

mmeybodi@aut.ac.ir

Table 1. Databases of German word trees

Structures	GN entries	CELEX entries	German trees
Flat	67,452	40,097	100,095
Deep-level	68,163	40,097	104,424
Merged with CELEX	68,171	n/a	100,986
Merged with CELEX Plus simplex words	68,171	n/a	112,086

5 Combining Morphological Databases with Segmenters

We combine the morphological database(s) with a morphological segmenter by a hybrid approach. Only if the database look-up fails, the time-consuming word splitter is invoked. See Fig. 4 for the combination with Moremorph/SMOR or Morphy.[8]

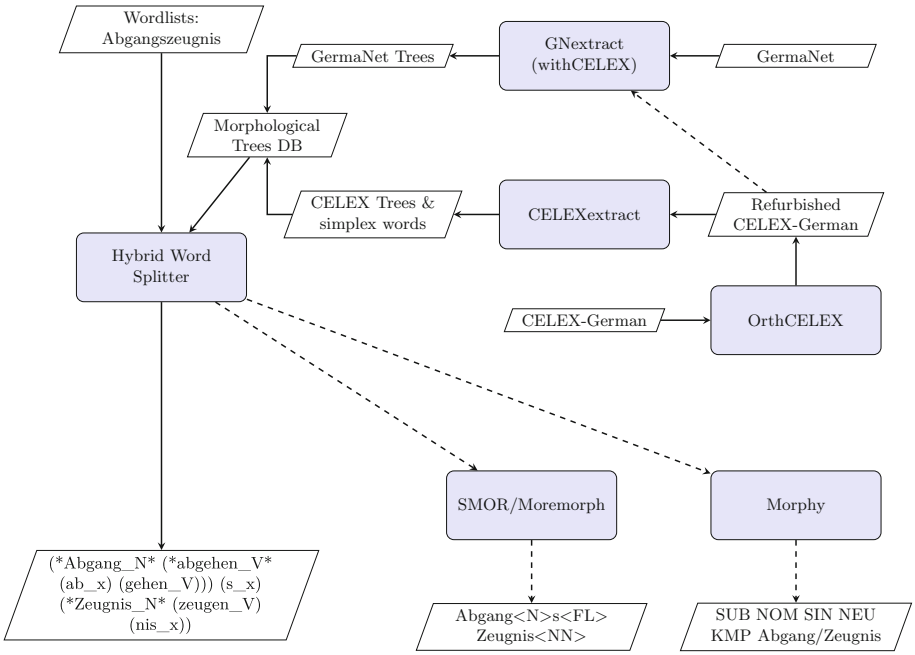


Fig. 4. Hybrid word analysis: morphological trees database and two different word segmenters as alternative methods for word splitting

[8] The scripts for the extraction of the morphological trees can be found online: https://github.com/petrasteiner/morphology.

mmeybodi@aut.ac.ir

6 Evaluation

For testing the performance, we are using *Korpus Magazin Lufthansa Bordbuch (MLD)* which is part of the DeReKo-2016-I [13] corpus[9]. It is an in-flight magazine with articles on traveling, consumption and aviation. For the tokenization, we enlarged and customized the tokenizer by [5] for our purposes. Multi-word units were automatically identified based on the multi-word dataset which we had augmented before. The resulting data comprises 276 texts with 5,202 paragraphs, 16,046 sentences and 260,115 tokens. The number of word-form types is 38,337. We are analyzing the lemmatized version of this corpus which was produced by the TreeTagger [25]. We add the simplex word forms of CELEX to the merged lexical database and use this database of morphological trees as first filter.

14,867 lemma types are not covered by the database, so they were re-analyzed by Morphy and SMOR/Moremorph. We manually checked the results of Moremorph and Morphy for the first 1,000 lemma types which could not be found in the database. Very often, these are rare or unusual words, so the output quality of both segmenters is much lower than usual. We then checked the correctness of the compound splitting.

7 Results

The details of the check against the database are included in Table 2, with a coverage of 49.29% for the lemma types and 60.59% for the lemma tokens. This direct lookup saves a lot of computational effort. According to the quality of the database, the recall is extremely close to these numbers.

The remaining 39.41% of all lemmas in text and 50.71% of all lemma types were analyzed in the following way:

We found that Morphy, with a somewhat limited lexicon (see Sect. 3.3), was able to process only 7,168 of the remaining lemma types, i.e. 51.79% of the lemma types were classified as unknown. But, with only 16 incorrect compound splits of 1,000, these results were of good quality. Due to an additional segmentation process, multi-word units were split to their parts, yielding a slightly higher number of lexical units (approx. 300). We get a coverage of 74.89%. For all lemma tokens, the newly retrieved ones comprise 83,582, therefore 241,117 of all lemmas inside the corpus could be recognized. This yields an overall coverage of 92.73%.

Moremorph, which calls SMOR with a more comprehensive lexicon (see Sect. 3.1), was able to process 13,461 lemmas (90.54%) of the words, the rest was classified as unknown. The number of analyzed lemma types (27,907) corresponds to a coverage of 95.20%.

The overall number of the lemma tokens which were covered by Moremorph amounts to 99,368. Adding this up to the number of words recognized by the

[9] See [16] and http://www1.ids-mannheim.de/kl/projekte/korpora/archiv/mld.html for further information.

mmeybodi@aut.ac.ir

database look-up, we get an overall coverage of 98.80% for correctly segmented words. We found 26 wrongly segmented words inside the sample of a thousand words which shows a good quality of the analysis.

Table 2. Coverage of Tree DBs

	Lemma types	Coverage	Lemma tokens	Coverage
Corpus size	29,313		260,014	
MergedDB + simplex	14,446	49.29%	157,535	60.59%
+ Morphy	21,953	74.89%	241,117	92.73%
+ Moremorph	27,907	95.20%	256,903	98.80%

8 Conclusion and Outlook

This paper demonstrates how updating and exploiting linguistic databases for morphological analyses can be performed. By simple look-up, we reached a recall of over 60% of the lemmas in text for the test corpus. As both databases were manually revised, we can speak of very reliable analyses. The remaining unanalyzed words can be mostly covered by conventional word segmenters. The results for the lemma types were a coverage of 76.91% for Morphy respectively 90.37% for Moremorph. These analyses have a flat structure. The results for the lemmas in texts are very promising: 92.73% respectively 98.80% of all words inside the texts were covered by the combined morphological analyses.

The direction of the future research is therefore straightforward: it will lead towards creating complex analyses out of existing ones and augmenting the lexical databases.

Acknowledgements. Work for this publication was partially supported by the German Research Foundation (DFG) under grant RU 1873/2-1 and by a Marie Curie Career Integration Grant within the 7th European Community Framework Programme. We especially thank Josef Ruppenhofer and Helmut Schmid for their constant assistance and cooperation, and Wolfgang Lezius for developing Morphy, for making it freely available and for the joint work.

References

1. Baayen, H., Piepenbrock, R., Gulikers, L.: The CELEX Lexical Database (CD-ROM). Linguistic Data Consortium, Philadelphia (1995)
2. Burnage, G.: CELEX: a guide for users. In: Baayen, H., Piepenbrock, R., Gulikers, L. (eds.) The CELEX Lexical Database (CD-ROM). Linguistic Data Consortium, Philadelphia (1995)

mmeybodi@aut.ac.ir

3. Cap, F.: Morphological processing of compounds for statistical machine translation. Ph.D. thesis, Universität Stuttgart (2014). https://doi.org/10.18419/opus-3474. http://elib.uni-stuttgart.de/opus/volltexte/2014/9768
4. Daiber, J., Quiroz, L., Wechsler, R., Frank, S.: Splitting compounds by semantic analogy. In: Proceedings of the 1st Deep Machine Translation Workshop, pp. 20–28. ÚFAL MFF UK (2015). http://aclweb.org/anthology/W15-5703
5. Dipper, S.: Tokenizer for German (2016). https://www.linguistics.rub.de/~dipper/resources/tokenizer.html
6. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
7. Geyken, A., Hanneforth, T.: TAGH: a complete morphology for German based on weighted finite state automata. In: Yli-Jyrä, A., Karttunen, L., Karhumäki, J. (eds.) Finite State Methods and Natural Language Processing, FSMNLP 2005. LNCS, vol. 4002, pp. 55–66. Springer, Berlin/Heidelberg (2006). https://doi.org/10.1007/11780885_7. http://www.dwds.de/static/publications/Geyken_Hanneforth_fsmnlp.pdf
8. Gulikers, L., Rattink, G., Piepenbrock, R.: German linguistic guide. In: Baayen, H., Piepenbrock, R., Gulikers, L. (eds.) The CELEX Lexical Database (CD-ROM). Linguistic Data Consortium, Philadelphia (1995)
9. Haapalainen, M., Majorin, A.: GERTWOL und morphologische Disambiguierung für das Deutsche. http://www2.lingsoft.fi/doc/gercg/NODALIDA-poster.html
10. Hamp, B., Feldweg, H.: GermaNet — a lexical-semantic net for German. In: Proceedings of ACL Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications, pp. 9–15 (1997). http://www.aclweb.org/anthology/W97-0802
11. Hanrieder, G.: MORPH — Ein modulares und robustes Morphologieprogramm für das Deutsche in Common Lisp. In: Hausser, R. (ed.) Linguistische Verifikation. Dokumentation zur Ersten Morpholymics 1994, pp. 53–66. Niemeyer, Tübingen (1996)
12. Henrich, V., Hinrichs, E.: Determining immediate constituents of compounds in GermaNet. In: 2011 Proceedings of the International Conference Recent Advances in Natural Language Processing, Hissar, Bulgaria, pp. 420–426. Association for Computational Linguistics (2011). http://www.aclweb.org/anthology/R11-1058
13. Institut für Deutsche Sprache: Deutsches Referenzkorpus/Archiv der Korpora geschriebener Gegenwartssprache 2016-I (2016). www.ids-mannheim.de/DeReKo. Release from 31 Mar 2016
14. Koehn, P., Knight, K.: Empirical methods for compound splitting. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary, 12–17 April 2003, vol. 1, pp. 187–193. Association for Computational Linguistics (2003). https://doi.org/10.3115/1067807.1067833. http://www.aclweb.org/anthology/E03-1076
15. Koskenniemi, K.: A general computational model for word-form recognition and production. In: 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics, Stanford University, California, 2–4 July 1984, pp. 178–181. Association for Computational Linguistics (1984). https://doi.org/10.3115/980491.980529. https://www.aclweb.org/anthology/P84-1038

mmeybodi@aut.ac.ir

16. Kupietz, M., Belica, C., Keibel, H., Witt, A.: The German reference corpus DeReKo: a primordial sample for linguistic research. In: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, Valletta, Malta, 17–23 May 2010, pp. 1848–1854. European Language Resources Association (ELRA) (2010). http://www.lrec-conf.org/proceedings/lrec2010/pdf/414_Paper.pdf
17. Lezius, W.: Morphologiesystem Morphy. In: Hausser, R. (ed.) Linguistische Verifikation. Dokumentation zur ersten Morpholympics 1994, pp. 25–35. Niemeyer, Tübingen (1996)
18. Lezius, W.: Morphy - German morphology, part-of-speech tagging and applications. In: Proceedings of the Ninth EURALEX International Congress, EURALEX 2000, Stuttgart, Germany, 8–12 August 2000, pp. 619–623 (2000). https://euralex.org/publications/morphy-german-morphology-part-of-speech-tagging-and-applications/
19. Lezius, W., Rapp, R., Wettler, M.: A morphology-system and part-of-speech tagger for German. In: Gibbon, D. (ed.) Natural Language Processing and Speech Technology, Results of the 3rd KONVENS Conference, pp. 369–378. Mouton de Gruyter (1996). https://arxiv.org/pdf/cmp-lg/9610006.pdf
20. Lezius, W., Rapp, R., Wettler, M.: A freely available morphological analyzer, disambiguator and context sensitive lemmatizer for German. In: Proceedings of the COLING-ACL 1998, Université de Montreal, Montreal, Quebec, Canada, 10–14 August 1998, vol. II, pp. 743–747 (1998). https://doi.org/10.3115/980691.980692. https://www.aclweb.org/anthology/P98-2123
21. Ma, J., Henrich, V., Hinrichs, E.: Letter sequence labeling for compound splitting. In: Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, Berlin, Germany, 16 August 2016, pp. 76–81. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/W16-2012. http://anthology.aclweb.org/W16-2012
22. Rapp, R., Lezius, W.: Statistische Wortartenannotierung für das Deutsche. Sprache und Datenverarbeitung **25**(2), 5–21 (2001)
23. Riedl, M., Biemann, C.: Unsupervised compound splitting with distributional semantics rivals supervised methods. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologie, San Diego, California, USA, 12–17 June 2016, pp. 617–622. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/N16-1075. http://www.aclweb.org/anthology/N16-1075
24. Schiller, A., Teufel, S., Thielen, C., Stöckert, C.: Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). Technical report, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung, and Seminar für Sprachwissenschaft, Universität Tübingen (1999). http://www.sfs.uni-tuebingen.de/resources/stts-1999.pdf
25. Schmid, H.: Improvements in part-of-speech tagging with an application to German. In: Armstrong, S., Church, K., Isabelle, P., Manzi, S., Tzoukermann, E., Yarowsky, D. (eds.) Natural Language Processing Using Very Large Corpora, pp. 13–25. Springer, Dordrecht (1999). https://doi.org/10.1007/978-94-017-2390-9_2
26. Schmid, H., Fitschen, A., Heid, U.: SMOR: a German computational morphology covering derivation, composition and inflection. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, Lisbon, Portugal, 26–28 May 2004. European Language Resources Association (ELRA) (2004). http://www.aclweb.org/anthology/L04-1275

mmeybodi@aut.ac.ir

27. Steiner, P.: Refurbishing a morphological database for German. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, 23–28 May 2016. European Language Resources Association (ELRA) (2016). https://www.aclweb.org/anthology/L16-1176
28. Steiner, P.: Merging the trees — building a morphological treebank for German from two resources. In: Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories, Prague, Czech Republic, 23–24 January 2018, pp. 146–160 (2017). https://aclweb.org/anthology/W17-7619
29. Steiner, P., Ruppenhofer, J.: Growing trees from morphs: towards data-driven morphological parsing. In: Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL 2015), University of Duisburg-Essen, Germany, 30 September–2 October 2015, pp. 49–57 (2015). https://gscl.org/content/GSCL2015/GSCL-201508.pdf
30. Steiner, P., Ruppenhofer, J.: Building a morphological treebank for German from a linguistic database. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018. European Language Resources Association (ELRA) (2018). https://www.aclweb.org/anthology/L18-1613
31. Sugisaki, K., Tuggener, D.: German compound splitting using the compound productivity of morphemes. In: 14th Conference on Natural Language Processing - KONVENS 2018, pp. 141–147. Austrian Academy of Sciences Press (2018). https://www.oeaw.ac.at/fileadmin/subsites/academiaecorpora/PDF/konvens18_16.pdf
32. Weller-Di Marco, M.: Simple compound splitting for German. In: Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017), Valencia, Spain, pp. 161–166. Association for Computational Linguistics (2017). https://doi.org/10.18653/v1/W17-1722. http://www.aclweb.org/anthology/W17-1722
33. Würzner, K., Hanneforth, T.: Parsing morphologically complex words. In: Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing, FSMNLP 2013, St. Andrews, Scotland, UK, 15–17 July 2013, pp. 39–43 (2013). https://www.aclweb.org/anthology/W13-1807
34. Ziering, P., Müller, S., van der Plas, L.: Top a splitter: using distributional semantics for improving compound splitting. In: Proceedings of the 12th Workshop on Multiword Expressions, Berlin, Germany, 11 August 2016, pp. 50–55. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/W16-1807. https://www.aclweb.org/anthology/W16-1807
35. Ziering, P., van der Plas, L.: Towards unsupervised and language-independent compound splitting using inflectional morphological transformations. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, USA, 12–17 June 2016, pp. 644–653. Association for Computational Linguistics (2016). https://www.aclweb.org/anthology/N16-1078

mmeybodi@aut.ac.ir



A Novel Topological Descriptor for ASL

Narges Mirehi, Maryam Tahmasbi(✉), and Alireza Tavakoli Targhi

Department of Computer Science, Shahid Beheshti University, G.C., Tehran, Iran
{n_mirehi,m_tahmasbi,a_tavakoli}@sbu.ac.ir

Abstract. Hand gesture recognition is a challenging problem in human computer interaction. A familiar category of this problem is American sign language recognition. In this paper, we study this problem from a topological point of view. We introduce a novel topological feature to capture and represent the shape properties of a hand gesture. The method is invariant to changes in rotation, scale, noise, and articulations. Due to the lack of ASL image database with all variations and signs, we introduced a database consisting of 520 images of 26 ASL gestures, with different rotation and deformations. Experimental results show that this algorithm can achieve a higher performance in comparison with state of the art methods.

Keywords: American sign language · Growing Neural Gas algorithm · Topological features · Adjacency matrix · Distance in graph · Boundary

1 Introduction

American sign language is communication tools between many normal people and deaf people. Automatic ASL recognition plays a significant role for people suffering from hearing issues. However, ASL recognition is a known difficult problem in computer vision due to the variety in shape, size, and direction of hand or fingers in different hand images [1]. Most previous researches extract relevant features and classify sign gestures using color-based and depth-based features [4,5,10]. ASL recognition without using sensor devices is a challenging problem due to the complexity of ASL gestures. However using sensor devices outside the laboratories is difficult for many reasons such as user inexperience, set up requirement and considerable costs [5,19]. So some of studies attempted to recognize ASL without using sensor devices [2,8,12,13,15,16]. [7] used wavelet decomposition features of hand images to recognize ASL problem. They applied neural networks to classify 24 static ASL alphabets but did not report the size of the dataset. Munib et al. [12] employed Cann's edge detection on 2D images and used Hough transform on the exterior and interior extracted edges to compute features. They classified only 14 ASL alphabets and some vocabularies and numbers based on neural network. Van den Bergh [18] proposed a method that recognized 6 hand gestures of a user. This method combined Haar wavelets features and neural network based on depth data and the RGB image. Stergiopoulou

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 274–289, 2020.
https://doi.org/10.1007/978-3-030-37309-2_22

mmeybodi@aut.ac.ir

et al. [16] used Growing Neural Gas algorithm to model hand region. They classified 31 different gestures by applying a likelihood-based technique. The drawback of their work is the inability to recognize gestures with finger's sticking together. Pugeault et al. [14] used Gabor filters as their hand shape features to recognize ASL images with depth data. They classified 24 ASL using a multi-class random forest. [4] used the depth and color data of image to extract the palm and finger regions of hand and computed geometrical properties such as the distances of the fingertips from the palm center, the curvature of the hand's contour and the shape of the palm region. They employed a multi-class SVM classifier to recognize 12 static American signs including digits and achieved 93.8% accuracy. Dahmani et al. [2] combined three shape descriptors: Tchebichef moments, Hu moments and geometric features. They evaluated their method on Arabic sign language alphabets and 10 ASL alphabets by SVM and KNN classifiers. The main limitations of geometric information based methods may be instability to rotation and articulation.

Sharma et al. [15] used contour trace features to describe hand shape and applied KNN and SVM classification techniques to classify 11 static alphabets of ASL with 76.82% accuracy. Dong et al. [5] used hand joint features to describe hand gesture and applied a random forest classifier to recognize 24 static ASL alphabets. Pattanaworapan et al. [13] divides ASL to fist and non-fist signs and used discrete wavelet transform to extract features of fist signs. To recognize non-fist signs, they divided hand image to $20 * 20$ or $10 * 10$ blocks and using coding table computed features. In a recent study, Ameen et al. [1] applied developed convolutional network to classify ASL using both intensities of image and depth data.

All pixel-based mentioned approaches have some limitations and suffer sensitivity to noise, articulation and some deformations. Since graphs are robust with respect to rotation and articulation, we use them to capture the topology of image. These graphs have limited number of vertices and make the size of the problem fixed in different scales. So, it can be used as powerful tools in shape recognition. In the previous study [11], the authors analyzed the ability of GNG graph to hand gesture recognition.

We use the Growing Neural Gas algorithm (GNG algorithm) introduced by Fritzke [6] to construct this graph. Two principal properties of this graph are low dimensionality and topological preserving. Then we extract the outer boundary of this graph that is a coarse estimation of the boundary of the object. After that, we compute the topological features by combining the geometric and graph theoretic features of this graph. Slight rotation and articulation are very natural in ASL gestures. Our method can easily handle these issues and achieve the recognition rate of 94.55% for non-fist signs, that is better than most recent studies. The rest of the paper is organized as follows: we summarize the basic definitions in Sect. 2. We construct the GNG graph, extract the outer boundary and define the topological features in Sect. 3. Then, we present algorithm and the results of American sign language gesture recognition and compare the results in Sect. 4. Finally, we present the conclusion in Sect. 5.

mmeybodi@aut.ac.ir

2 Basic Definitions

In this section, we review some primary definitions from graph theory. Most of the definitions and results can be found in graph theory text books. Let $G = (V, E)$ be a graph with $V = \{1, 2, ..., n\}$. The *adjacency matrix* of G is an $n \times n$, 0–1 matrix $A_G := [a_{ij}]$, where $a_{ij} = 1$ if and only if ij is an edge. A *walk* in a graph G is a sequence $W := v_0, v_1 ... v_{l-1}, v_l$, of vertices of G such that there is an edge between every two consecutive vertices. The length of this walk is l. If A is the adjacency matrix of a graph, the ij-th entry in A^k is the number of walks of length k between i and j in G. A *path* is a walk with no repeated vertices.

3 Our Method

Given a binary image or the silhouette of an object, we extract topological features. To gain robustness to scale, we use a mesh to fill inside the image. This mesh is a graph with a fixed number of vertices that are placed almost uniformly inside the image. Different approaches can be used to construct this graph. We use a known method based on growing neural gas algorithm (GNG algorithm) [6] and call it the *GNG graph*. This graph is not sensitive to the existence of small holes inside the image. It is also robust with respect to narrow and small noises on the boundary. The main steps of our method include:

1. Estimating the image with a GNG graph whose vertices are distributed almost uniformly inside the image.
2. Extracting the outer boundary of the GNG graph, using computational geometry approaches.
3. Identifying peaks and troughs on the boundary of the image (boundary features) using a combination of geometrical and topological approaches

In the rest of this section, we describe each step separately, providing more detail.

3.1 Computing the GNG Graph

Growing Neural Gas algorithm (GNG) is an incremental network which learns the topology [6]. The GNG algorithm constructs a low-dimensional subspace of the input data and describes the topological structure of it as well. The algorithm constructs a graph whose vertices are uniformly distributed inside the image. We experimented GNG graph with the various number of neurons from 100, 150, 200, 250, 300 and observed that 200 neurons are sufficient. Figure 1a shows an example.

mmeybodi@aut.ac.ir

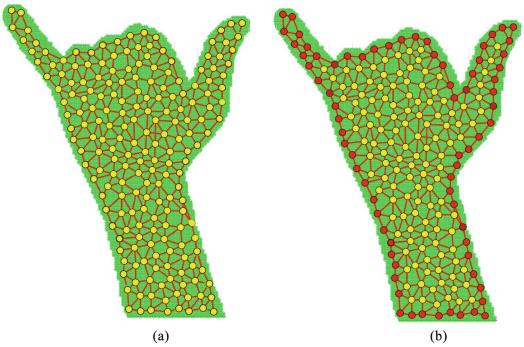


Fig. 1. (a) A GNG graph of an hand image (b) The vertices on the outer boundary are shown in red.

3.2 Extracting the Outer Boundary of the GNG Graph

As mentioned, the GNG graph is a geometric graph, i.e. each vertex has a coordinate and each edge is a line segment. To extract the boundary, we use the idea of convex hull algorithms [3]. We find the leftmost vertex v and its neighbor u with the smallest clockwise angle with the upward vertical half-line starting at v, and insert v and u in C. Then we walk around the boundary and add new vertices to C. In each step, we consider the two last vertices u_{i-1} and u_i in C and for all vertices v, adjacent to u_i, we compute the size of the clockwise angle at u_i between the edges u_i, u_{i-1} and u_i, v, the vertex with minimum angle is the next vertex on the boundary and is inserted in C. We repeat the above step until the walk is closed [11]. Figure 2b shows an example.

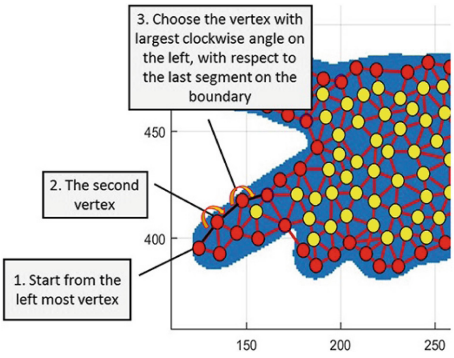


Fig. 2. The outer boundary extraction, The vertices on the outer boundary are shown in red.

mmeybodi@aut.ac.ir

3.3 Bulges

Peaks and troughs on the boundary of an image show the shape of it. We define the concept of a *bulge* to show a peak on the boundary of an image.

Let G be a graph, and H be the graph (cycle) representing the outer boundary of G. Suppose that the vertices of H are named $v_1, v_2, \ldots, v_k$ in clockwise order of appearance on the boundary.

Definition 1. *Given a constant $c > 1$, let u_i and u_j, $(i < j)$ be two vertices of H such that $d_H(u_i, u_j) \geq c \times d_G(u_i, u_j)$. We call the pair (u_i, u_j) a c-pair. A path between u_i and u_j in H is called H-path and the shortest path between u_i and u_j in G is called G-path. Figure 4(a) shows an example of H-path and G-path. Two c-pairs are intersecting, if their H-paths have common vertices, except u_i and u_j.*
If (u_i, u_j) and (u_k, u_l) are two intersecting c-pairs, then the union of these c-pairs is a pair (u_r, u_s) where $r = \min\{i, k\}$ and $s = \max\{j, l\}$. Note that the union of two c-pairs is not necessarily a c-pair. Let (u_i, u_j) be the union of all intersecting c-pairs. The subgraph graph consisting of the H-path between u_i and u_j, the shortest path between them in G and all vertices and edges between these paths is called a bulge. The vertices u_i and u_j are called the basic vertices.

The parameter c is determined with respect to the application. Smaller values of c make the shape more sensitive to noise, and larger values ignore small bulges. Figure 4(a) shows an example of a bulge. In this figure, the edges of H-path between these vertices are black and the edges of G-path are white.

The diagram in Fig. 3 classifies topological features of an object . In the following, we describe these features with more detail.

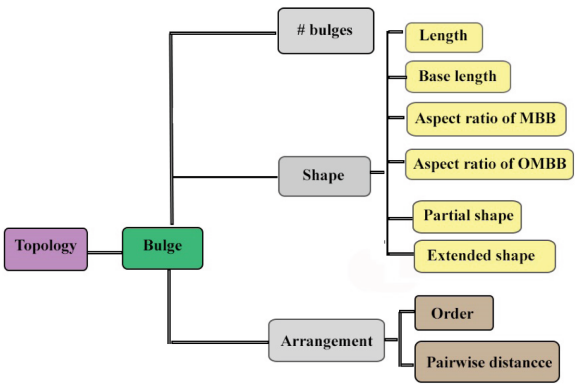


Fig. 3. Topological features of a shape.

1. ♯ **Bulges.** This feature shows the number of the bulges. When the shape has no bulges, we suppose that the whole image is one bulge. In this case, there

mmeybodi@aut.ac.ir

is no basic vertices, H-path or G-path. The only features extracted in this case are 2c and 2d.

2. **Shape.** The following properties of a bulge help in describing the shape of it.
   (a) **Length.** This feature measures the length of the H-path between the basic vertices of the bulge. Since all vertices are uniformly distributed inside the image, this feature is independent of scale. Figure 4(a) shows a bulge with length 14.
   (b) **Base length.** This feature measures the length of G-path between basic vertices. For example, the base length of the bulge shown in Fig. 4(a) is 2.
   (c) **Aspect ratio of MBB.** This feature shows the elongation of the bulge and is defined as the aspect ratio of its MBB. The MBB of a bulge is shown in Fig. 4(b) with a dashed rectangle.
   (d) **Aspect ratio of OMBB.** In case of rotation, the bounding box of image changes, but OMBB does not. So, the aspect ratio of OMBB is used as a topological feature. (see Fig. 4(b)).
   (e) **Partial shape.** In order to recognize the shape of a bulge, we can recognize the shape of some parts of its boundary. Here we use one-third of the vertices in the middle of H-path and compute their OMBB. If the middle of a bulge is flat, the aspect ratio is small. Figure 5 shows two examples.
   (f) **Extended shape.** In order to keep the shape of troughs around a peak (bulge), we extend the bulge and compare its OMBB and the extended bulge. If u_i and u_j are the basic vertices of a bulge, we extend the bulge by adding vertices u_{i-k} and u_{j+k} $k \in \{1, 2, 3\}$ to it. An example is shown in Fig. 4(c).
3. **Arrangement.** The arrangement of bulges contains valuable information about the shape and is used to recognize different shapes with similar bulges. Three features help to recognize the arrangement:
   (a) **Order.** Bulges are stored in clockwise order of appearance on the boundary.
   (b) **Pairwise distance.** It is the length of the shortest H- path between basic vertices of two bulges.

3.4 Extracting Bulges

All our GNG graphs have 200 vertices. Different methods can be applied to determine the width and length of fingers. There is a standard measurement for hand presented in [9]. According to this, middle finger length (fingertip to knuckle) is 5.5 times finger width and the length of little finger is not shorter than half of middle finger. Also, experimental study shows that the base length of a bulge representing one finger is at most 2. On the other hand, each finger is a long and narrow bulge, so, the distance of the basic vertices in H, must be at least 5. So, the parameter c mentioned in definition of bulge (Definition 1) equals 2.5. c-pairs with $c = 2.5$, are candidates for bulges representing a single finger,

mmeybodi@aut.ac.ir

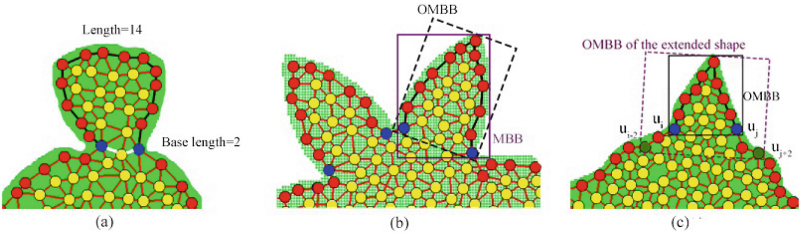


Fig. 4. (a) H-path (black arcs between blue vertices) and G-path (white arcs between blue vertices) of a bulge are shown, (b) MBB (solid line) and OMBB (dashed line) of a bulge are drawn, (c) OMBB of a bulge (solid) and OMBB of the extended shape (dashed) are shown.

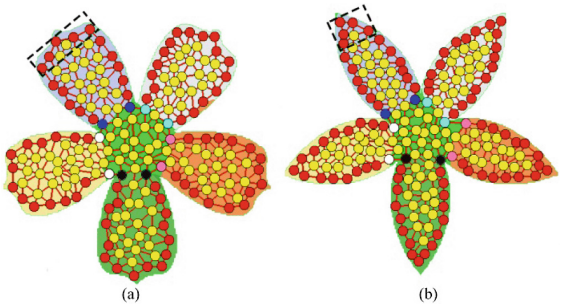


Fig. 5. (a) and (b) show two flowers including the same number of bulges while their bulges have different partial shape (dashed rectangle).

two or three fingers sticking together or the wrist. c-pairs with $d_G(u_i, u_j) = 2$ are appropriate candidates for a single finger and c-pairs with $d_G(u_i, u_j) = 4$ are the candidates for sticking fingers (the sticking fingers have about twice the width of a single finger). The wrist is another bulge that has a significant role in recognizing a gesture. c-pairs with $d_G(u_i, u_j) \in \{5, 6, 7\}$ are candidates for bulges representing the wrist.

We measure all distances, as distances in graph. Since the vertices are distributed almost uniformly inside the silhouette, this is a fair approximation of distance and is not sensitive to rotation, articulation and scale.

The matrix $A-B$ shows the edges of G that are not in H, so, $(A-B)^k$ shows the number of walks avoiding H with length k between pairs of vertices. The candidate c-pairs for single fingers are pairs (i, j) such that $(A-B)^2[i, j] \neq 0$. We also need to enforce the condition that distance between these vertices in H is at least 5, i.e. the corresponding entry in $B^3 + B^4$ must be zero. So, these candidates c-pairs are the pairs of vertices that their corresponding entry in $C = ((A-B)^2 > 0) - ((B^3 + B^4 > 0)$ is non zero. The matrix $(A-B)^2 > 0$ is a binary matrix where each non zero entry of $(A-B)^2$ equals 1. So, the matrix C is a binary matrix.

mmeybodi@aut.ac.ir

We use the matrix $((A-B)^3 > 0) - (\sum_{n=4}^{6} B^n) > 0$ and $((A-B)^4 > 0) - (\sum_{n=5}^{8} B^n) > 0$ for finding sticking fingers.

We also compute the basic vertices of the bulge corresponding to the wrist in a similar way. We suppose that basic vertices of the wrist have distance of length 5, 6 or 7 in G but their distance in H is more than 11, so, the matrices

$$((A-B)^k > 0) - (\sum_{n=k+1}^{11} B^n) > 0, k \in \{5,6,7\}$$

are used for finding the wrist. There is an important detail in finding the wrist, the boundary of the wrist must not contain any fingers. Enforcing this condition helps removing dummy bulges in the corners of the wrist.

4 American Sign Language (ASL)

ASL sign gestures are divided into fist and non-fist sign gestures. Figure 6 shows these two groups. The letters J and Z have motions, so are usually ignored in hand gesture recognition approaches. The silhouette of fist sign gestures includes shapes with similar topology and are not distinguishable from topological point of view. So, we apply our method to recognize the non-fist signs $\{B, C, D, F, G, H, I, K, L, P, Q, R, U, V, W, X, Y\}$. Signs G and Q are classified in the same class due to similarity of topology of their silhouette.

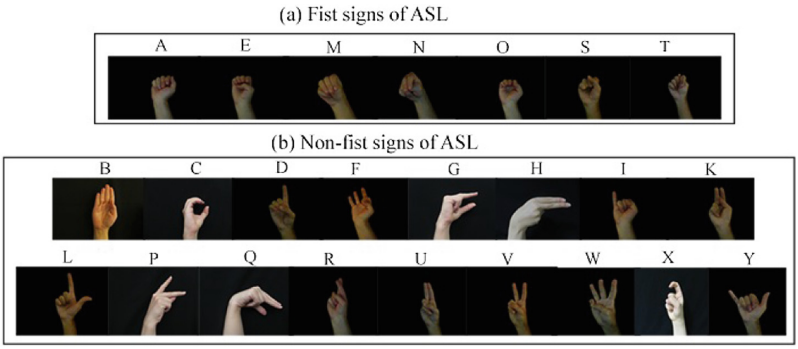


Fig. 6. (a) Fist (b) non-fist sign gestures of ASL

A number of image databases were introduced for benchmarks on hand signs. But most of them are incomplete and just include few gestures or used sensor devices. We provided a new database of 26 American sign hand gestures from right hand where the palm of the hand facing the camera. It includes 520 images from 26 gestures. We call this database, SBU-ASL-1[1]. Figure 6 shows

[1] It is available at http://facultymembers.sbu.ac.ir/tahmasbi/index.php/en/.

mmeybodi@aut.ac.ir

Table 1. Topological features used in recognizing ASL alphabet. Gestures are classified according to the number of bulges.

ASL Classification		
♯ bulges	ASL signs	Topological separating features
1	B, fist signs	– Aspect ratio of OMBB
2	D, H, I, R, U, X	– Base length (1 finger or sticking finger)(D, X, I, from H, U, R) – Pairwise distance from wrist – MBB – OMBB – OMBB of partial shape – OMBB of extended shape
3	C, G, K, L, P, Q, V, Y	– Length – Pairwise distance from wrist – Aspect ratio of OMBB – Aspect ratio of MBB – OMBB of extended shape
4	W, F	– Pairwise distance from wrist

some images of SBU-ASL-1. These images are colored with a black background in different sizes.

Table 1 shows the topological features used in recognizing different sign gestures in ASL. In this paper, data has been collected from the database, SBU-ASL-1.

4.1 Topological Features

The images are divided into 4 classes based on the number of their bulges (see Table 1). The wrist is considered as first bulge and fingers are sorted in clockwise order from little finger to thumb (if available). At the first step, we classify the gestures by the number of bulges, then we use the different defined features (according Table 1) to separate the gestures with the same number of bulges. Let $D_1, D_2, D_3, ...D_n$ show the distance between consecutive bulges. The parameters $Ratio$, $Ratio1$, $Ratio2$ and $Ratio3$ for a bulge are defined below and are used to separate sign gestures in the same class.

$Ratio = A\ aspect\ ratio\ of\ MBB$
$Ratio1 = A\ aspect\ ratio\ of\ OMBB$
$Ratio2 = A\ aspect\ ratio\ of\ OMBB\ of\ partial\ shape$
$Ratio3 = A\ aspect\ ratio\ of\ \ OMBB\ of\ extended\ shape$

Now we present our algorithm for sign recognition in each class.

mmeybodi@aut.ac.ir

Gestures with One Bulge: For sign gestures with only one bulge, that is the wrist, we ignore the wrist and compute the aspect ratio of the rest of image and use it for gesture recognition.

Gestures with Two Bulges: Sign gestures $\{D, H, I, R, U, X\}$ contain two bulges: one bulge for the wrist and another for the finger. In these gestures, we omit the wrist again and examine the remaining bulge for gesture recognition. The base length of signs U and H are 3 or 4, while the base length of other signs is 2. So, U and H are simply separated from D, I, R and X, comparing the base length of their bulge. The sign R has two raised fingers, but since they cover each other partially, the corresponding bulge has base length 2.

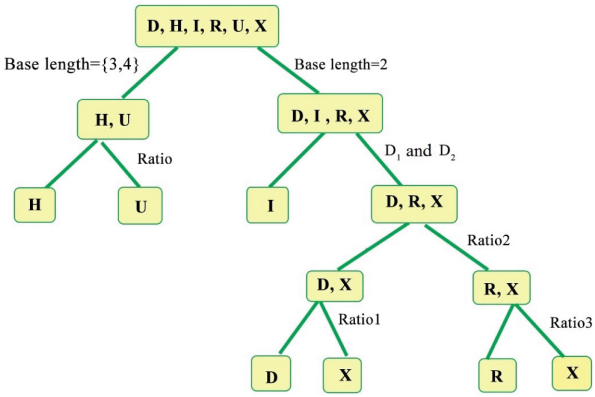


Fig. 7. Topological features separating letters in signs with only one finger.

For separating D, R, I, X, we use the following facts:

1. In I, $D_1 < D_2$, and for all other signs $D_1 \geq D_2$.
2. The sign gestures D, R and X are separated with comparing parameters *Ratio*1, *Ratio*2 and *Ratio*3. The value of *Ratio*2 for D and R has a significant difference. So, the image is either D and X, or R and X.
3. The partial shape of D is triangle-shaped, while the partial shape of R is flat, the aspect ratio of OMBB of partial shape can help in separating them.
4. In sign D, the index finger is raised while in sign X, it is bent. So, we use the aspect ratio of OMBB to separate them.
5. The shape of troughs around R, X are different. So, the aspect ratio of extended shape can help separating the signs.

Also, since the signs U and H are similar and differ in the direction of fingers, they are separated using the aspect ratio of the MBB of the bulge. Figure 7 shows the features used in separating these signs.

mmeybodi@aut.ac.ir

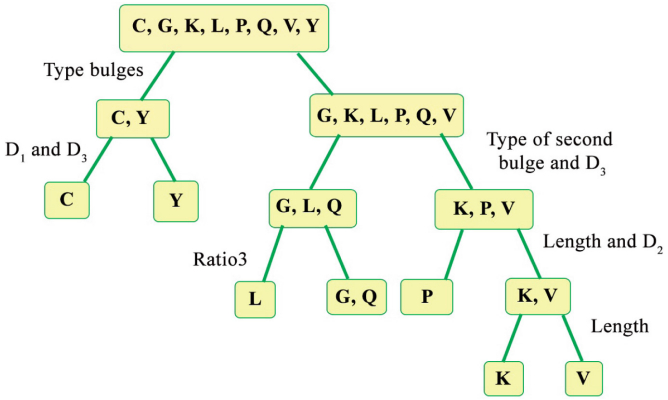


Fig. 8. Different topological features used in recognizing gestures with three bulges.

Gestures with Three Bulges: Signs $\{C, G, K, L, P, Q, V, Y\}$ include three bulges: one wrist and two fingers. The first step is to identify the type of first and last finger by considering the pairwise distance from wrist. Then topological features of these bulges are computed and compared. To separate signs in this class, we use the following facts:

1. The silhouette of sign gestures K, V and P are similar, but still, they are different. The significant difference between K and V appears in the length of their bulges.
2. In sign K, thumb is placed between index and middle fingers, therefore the length of bulges corresponding to these fingers is shorter than bulges in V.
3. The sign P is separated from K and V by comparing the length of bulges and distance between them. In sign P, the second bulge has the shorter length than the first bulge and the distance between theses bulges is more than that in K and V.
4. In both signs Y and C, the second finger is thumb and first finger is little; however, in Y, the difference between D_1 and D_3 is more than that in C.
5. In signs L, G, and Q, the second finger is thumb and the distance between fingers is more than 2. The fingers are closer to each other in G and Q than in L. So, Ratio3 separates G and Q from L. G and Q are considered in the same class since they are the same from topological point of view.

Figure 8 shows features used in recognizing each sign in this class.

Gestures with Four Bulges: The signs W and F contain four bulges, one wrist, and three fingers. These signs differ in finger type and, in fact, pairwise distance from the wrist. If $D_1 < D_4$ the sign gesture is F otherwise it be W.

mmeybodi@aut.ac.ir

4.2 The Result of Experimental Study on SBU-1

We classify ASL alphabet to two group; fist signs and non-fist signs. If the number of extracted bulges in a hand image is 1, the sign gesture might be B or a non-fist sign. Sign B is simply separated from non-fist signs by computing the aspect ratio of OMBB. Accordingly fist signs and non-fist signs are separated from each other. Table 2 shows our sign grouping accuracy. We can recognize non-fist signs from fist signs with accuracy 100%. Afterward, we recognize non-fist ASL gestures.

Table 2. Sign grouping performance

	Recognition performance	
	Fist signs	Non-fist signs
Fist signs	97.24	2.98
Non-fist signs	0	100

Table 3. Confusion matrix of non-fist signs

	B	C	D	F	G	H	I	K	L	P	R	U	V	W	X	Y
B	**95**	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0
C	0	**96**	1	0	0	0	0	0	1	2	0	0	0	0	0	0
D	0	0	**95**	0	0	0	0	0	0	0	5	0	0	0	0	0
F	0	0	0	**99**	0	0	0	0	0	0	0	0	0	1	0	0
G	0	0	0	0	**91**	0	0	0	2	7	0	0	0	0	0	0
H	1	0	2	0	0	**96**	0	0	0	0	0	0	0	0	1	0
I	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	**91**	0	8	0	0	1	0	0	0
L	0	1	0	0	7	0	0	0	**92**	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	5	0	**95**	0	0	0	0	0	0
R	0	0	1	0	0	0	0	0	0	0	**90**	0	0	0	9	0
U	4	0	0	0	0	0	0	0	0	0	1	**95**	0	0	0	0
V	0	0	0	0	3	0	0	6	0	2	0	0	**89**	0	0	0
W	0	0	0	2	0	0	0	0	0	0	0	0	0	**98**	0	0
X	0	0	5	0	0	0	0	0	0	0	1	0	0	0	**94**	0
Y	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	**97**

Table 3 shows the confusion matrix of our method. The diagonal elements show the accuracy of correct recognition in each sign. We succeeded to recognize

mmeybodi@aut.ac.ir

non-fist signs with average 94.55% accuracy. The best recognition rate is for I with 100% while the weakest recognition rate is for sign R and V with 90%.

4.3 Comparing the Results

In this section, we compare our method with some similar approaches and report the results. Some recent researches on ASL are based on using a Microsoft Kinect device [1,5,14,15]. Since Kinect devices include many significant limitations, studying ASL recognition without using Kinect devices is worthwhile.

Table 4 shows the comparison of recognition accuracy between previous studies and our method [1,5,12–15].

Our method recognized non-fist alphabets on dataset SBU-ASL-1 with overall accuracy 94.55% which is better performance than [13]. We recognize signs C, I, L, U, W, X with the highest rate. These studies recognized only some non-fist signs or used sensor device. [12] could recognize 11 non-fist ASL alphabets with overall accuracy of %89.33, while their data set contained only 15 images from any sign gesture. [15] used a contour tracing descriptor to recognize 7 alphabets of ASL and recognized the non-fist signs C, F, G, H, I, R with overall accuracy of %77.9. [14] applied Gabor filters on image and depth data which is collected with Kinect device. They could recognize non-fist signs with overall accuracy 61.18%. [13] divided hand image to $20*20$ or $10*10$ blocks then using ANN recognized 16 alphabets of non-fist signs with %89.38 accuracy. This method considers alphabet K and V in the same class and does not separate them because of silhouette similarity. Also [5] used depth data and computed hand joint angles to describe the hand gesture and recognized non-fist alphabets with overall accuracy 85.81%. [1] achieved overall accuracy 87% on non-fist signs using a CNN method on the images with depth data. Some other previous methods by [2,8,17] tested their approach on alphabets A, B, C, D, G, H, I, L, V, Y based on data base [17] with overall %91.8, %93.1, %95.2 accuracy respectively. Triesch data base contains only 10 signs from ASL alphabet and does not include some complicated signs.

4.4 Analysis Properties of Our Method

In this paper, GNG graphs are applied with the limited number of vertices so, it makes the size of the problem fixed in different scales. Also, the properties of graphs do not change by rotation and articulation. SBU-ASL-1 contains images with various rotation, articulation, and scale. Experimental results on SBU-ASL-1 show that our method is robust to rotation, articulation, and scale.

The noise is an unavoidable and challenging problem in hand gesture recognition. Most current methods are based on the local properties of pixels so the destruction of pixels reduces their performance considerably. For instance, skeleton-based methods cannot tolerate noise effects in object's boundary and their stability decreases in occupation of noise. An interesting characteristic of our method is stability against noise. To show the ability of our method against noise, we added Gaussian noises to a hand gesture. Gaussian noises with zero mean and standard deviation σ as 0.5, 1, 1.5 and 2 are added to all pixels on in both x and y directions. The noise is increased when parameter σ is increased.

mmeybodi@aut.ac.ir

Table 4. Non-fist sign recognition comparison. Star symbols mention to studies used sensor device.

Input sign				Recognition performance				
	[12]	[14]*	[15]	$10 * 10$ blocks [13]	$20 * 20$ blocks [13]	[5]*	[1]*	Our method
B	–	83	93.3	86	98	88	94	**94.2**
C	–	57	90	80	62	90	78	**94.2**
D	100	37	–	98	90	93	86	94.2
F	100	35	75	100	88	90	97	99.2
G	–	60	65	92	88	73	90	**92.5**
H	–	80	75	96	88	82	83	**95.8**
I	80	73	85	88	98	90	93	**100**
K	100	43	95	–	–	73	81	91.7
L	86.7	87	–	86	90	95	96	92.5
P	–	57	–	100	82	69	72	**95**
R	86.7	63	75	88	84	82	81	**90**
U	86.7	67	–	80	94	95	82	95
V	93.3	87	–	88	96	88	87	90
W	93.3	53	–	86	92	89	97	**98.4**
X	–	20	–	92	94	84	83	**93.4**
Y	73.3	77	–	100	96	92	92	**96.7**
Average	89.33	61.18	77.5	87.38	89.38	85.81	87	**94.55**

Figure 9 shows noisy images with different Gaussian noises. We observe that the increasing value of noise has not considerable effect on our method. Extracting boundary of noisy objects is a challenging problem in computer vision. Our method extracts the boundary of the GNG graph and this graph is stable against noise.

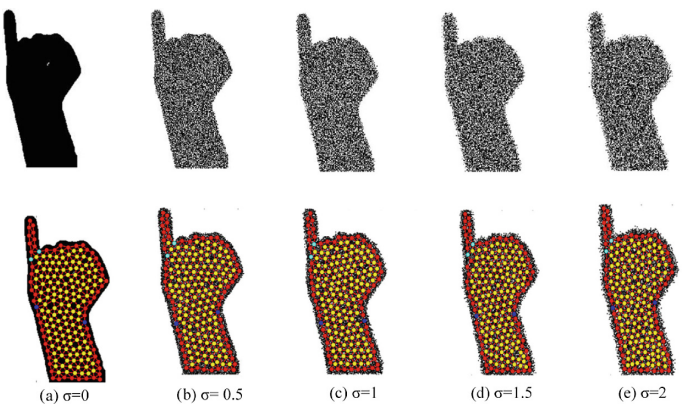


Fig. 9. Images with different Gaussian noises σ and their GNG graph.

mmeybodi@aut.ac.ir

5 Conclusion

In this paper, we defined a new graph-based method to ASL recognition with significant topological features. We use a GNG graph to extract topological features. This graph is not sensitive to noise and perturbation of the boundary, rotation, scale, and articulation of the image. This approach considers the topological features of the boundary like peaks and troughs, bounding boxes, convex hulls and ignores the geometrical features, like size, angle, Euclidean distance, and slope to generate shape features that are invariant to rotation, scale, articulation, and noise. Both region and boundary of an image are used for extracting topological features so the proposed method dose not include the limitation contour based methods. We could achieve the recognition rate of 94.55% for non-fist sign gestures.

References

1. Ameen, S., Vadera, S.: A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images. Expert Syst. **34**(3), e12197 (2017)
2. Dahmani, D., Larabi, S.: User-independent system for sign language finger spelling recognition. J. Vis. Commun. Image Represent. **25**(5), 1240–1250 (2014)
3. De Berg, M., Van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational geometry. In: Computational Geometry, pp. 1–17. Springer, Heidelberg (1997)
4. Dominio, F., Donadeo, M., Zanuttigh, P.: Combining multiple depth-based descriptors for hand gesture recognition. Pattern Recogn. Lett. **50**, 101–111 (2014)
5. Dong, C., Leu, M.C., Yin, Z.: American sign language alphabet recognition using Microsoft Kinect. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 44–52 (2015)
6. Fritzke, B.: A growing neural gas network learns topologies. In: Advances in Neural Information Processing Systems, pp. 625–632 (1995)
7. Isaacs, J., Foo, S.: Hand pose estimation for American sign language recognition. In: Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory, pp. 132–136. IEEE (2004)
8. Kelly, D., McDonald, J., Markham, C.: A person independent system for recognition of hand postures used in sign language. Pattern Recogn. Lett. **31**(11), 1359–1368 (2010)
9. Klein, H.A.: The Science of Measurement: A Historical Survey. Courier Corporation, Chelmsford (2012)
10. Li, Y., Wang, X., Liu, W., Feng, B.: Deep attention network for joint hand gesture localization and recognition using static RGB-D images. Inf. Sci. **441**, 66–78 (2018)
11. Mirehi, N., Tahmasbi, M., Targhi, A.T.: Hand gesture recognition using topological features. Multimed. Tools Appl. **78**, 1–26 (2019)
12. Munib, Q., Habeeb, M., Takruri, B., Al-Malik, H.A.: American sign language (ASL) recognition based on hough transform and neural networks. Expert Syst. Appl. **32**, 24–37 (2007)
13. Pattanaworapan, K., Chamnongthai, K., Guo, J.M.: Signer-independence finger alphabet recognition using discrete wavelet transform and area level run lengths. J. Vis. Commun. Image Represent. **38**, 658–677 (2016)

mmeybodi@aut.ac.ir

14. Pugeault, N., Bowden, R.: Spelling it out: real-time ASL fingerspelling recognition. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1114–1119. IEEE (2011)
15. Sharma, R., Nemani, Y., Kumar, S., Kane, L., Khanna, P.: Recognition of single handed sign language gestures using contour tracing descriptor. In: Proceedings of the World Congress on Engineering, pp. 3–5 (2013)
16. Stergiopoulou, E., Papamarkos, N.: Hand gesture recognition using a neural network shape fitting technique. Eng. Appl. Artif. Intell. **22**, 1141–1158 (2009)
17. Triesch, J., von der Malsburg, C.: Classification of hand postures against complex backgrounds using elastic graph matching. Image Vis. Comput. **20**, 937–943 (2002)
18. Van den Bergh, M., Van Gool, L.: Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In: 2011 IEEE Workshop on Applications of Computer Vision (WACV), pp. 66–72. IEEE, January 2011
19. Wang, C., Liu, Z., Chan, S.C.: Superpixel-based hand gesture recognition with kinect depth camera. IEEE Trans. Multimed. **17**(1), 29–39 (2015)

mmeybodi@aut.ac.ir



Pairwise Conditional Random Fields for Protein Function Prediction

Omid Abbaszadeh and Ali Reza Khanteymoori(✉)

Computer Engineering Department, University of Zanjan, Zanjan, Iran
{o.abbaszadeh, khanteymoori}@znu.ac.ir

Abstract. Protein Function Prediction (PFP) is considered one of the complex computational problems where any protein can simultaneously belong to more than one class. This issue is known as Multi-label classification problem in pattern recognition. Multi-label data sets are kind of data where each instance belongs to more than one class. This feature differentiates multi-label classification from the standard types of data classification. One of the challenges in multi-label classification data is correlation between the labels. This feature makes the issue cannot be classified into distinct sets of classification divided. Another major challenge is the high dimensional data in some applications. This paper presents a new method for the Protein Function Prediction and classification of multi-label data using conditional random fields. More specifically, the proposed approach is a method based on Pairwise Conditional Random Fields which considered the relationship of the labels. After introducing the Pairwise Conditional Random Fields optimization problem and solving it, the proposed method is evaluated under different criteria and the results confirm higher performance compared to available multi-label classifiers.

Keywords: Protein Function Prediction (PFP) · Multi-label classification · Conditional Random Fields · Pairwise Conditional Random Fields

1 Introduction

Protein sequences identification in some of the organisms, such as humans, is leading to a new era in the biology and related sciences. The main goal in this field is identification the sequence and structure of the countless proteins which are fully recognized, but detailed information on function is not available [1]. The first approach to identify proteins function is laboratory methods. These methods are very expensive and time-consuming. Therefore the uses of computational methods are a good choice. Among the available computational methods, machine learning techniques are well placed to solve this problem. In fact, this technique of using existing data sources, learning the model that this model is able to predict the function of an unknown protein. The similar issue of Protein Function Prediction (PFP) is Multi-label Classification (MLC) in machine learning and pattern recognition. In traditional data classification, each sample

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 290–298, 2020.
https://doi.org/10.1007/978-3-030-37309-2_23

mmeybodi@aut.ac.ir

is associated with one label but in MLC, each sample is associated with more than one label. This is what we see in the function of the proteins that each protein can have different and multiple functions.

Generally, different methods to PFP or MLC divided into two categories: (1) Data-level (2) Algorithm level. In data-level methods, the data set is split into multiple single label datasets. This approach can be based on labels or instances. The main drawback of these methods is their high time complexity in the large datasets. In algorithm level methods, data classification is performed by changing the conventional classification algorithms [2]. Due to low complexity and well scalability, these methods are suitable for large data such as protein function datasets.

In this paper, we introduced Pairwise Conditional Random Field (Pairwise CRF) to PFP or MLC that is an algorithm level method. Conditional random fields (CRFs) are a probabilistic model for classifying structured data, such as protein sequences. The main idea in CRF is that of defining a conditional probability distribution over the label data given a particular observation data, rather than a joint distribution over both label and observation data. The main advantage of CRFs over Markov Random Fields (MRFs) is their conditional nature. Pairwise CRF is a type of the CRFs that the relationship between the labels is applied in the model. Generally, exact inference in MRFs and CRFs is a NP-hard problem [3]. Many approximate and optimization algorithms have been proposed for inference in the CRFs. Scalability and correlation between the labels have not been met in most approach as well.

The most important step in CRF and Pairwise CRF is determining the parameters of the model. In this paper, we used the log-likelihood function then present an optimization method to obtain model parameters then solving it using the Frank-Wolfe algorithm. The experimental results showed the advantage of the proposed method on standard datasets under different criteria.

The remainder of this paper is organized as follows. In Sect. 2, we describe related works. Section 3 describes the proposed method. Section 4 describes the data sets used in our experiments and shows the results on different metrics. Section 5 discusses the conclusions we reached based on these experiments and outlines directions for future research.

2 Related Work

MLC tasks are everywhere in real-world problems. For instance, document categorization, image processing, gene prediction and PFP. Numerous algorithms have been proposed for MLC problem and each of these algorithms can be used to PFP. Ensemble methods, Support vector machines (SVM), Decision Trees (DT) and lazy learners such as k-Nearest Neighbors (kNN) are the most popular classifiers which can used to PFP.

Yu et al. [4] developed a graph-based transductive learner for PFP and called it TMC (Transductive Multi-label Classifier). The ensemble of TMCs by integrating multiple data sources train a directed bi-relation graph for each base classifier. The RAndom k-labELsets (RAkEL), developed by Tsoumakas and

mmeybodi@aut.ac.ir

Katakis [5], transforms MLC task into multiple binary classification problems. RAkEL creates a new class for each subset of labels and then train classifiers on the random subset of labels.

Support Vector Machine proposed by Vapnik in 1992 is another most interesting classifier in pattern recognition. Elisseeff et al. [6] proposed Multi-label SVM and Rank-SVM that incorporates a ranking loss within the minimization function.

C4.5 decision tree is the well-known algorithm for single data classification. Multi-Label C4.5 (ML-C4.5) [7] is an adaptation of the C4.5 algorithm for multi-label classification by the multiple labels in the leaves of the C4.5 trees. C4.5 uses entropy formula for selecting the best split. Clare et al. [7] modified the formula for calculating entropy for solving MLC. ML-C4.5 uses the sum of entropies of the class variables.

Multi-label kNN (ML-kNN) [8] is an extension of the popular k-nearest neighbors (kNN) algorithm. In this approach, for each test sample, its k-nearest neighbors in the training set are identified and based on statistical information obtained from the labels of these neighboring samples, the maximum a posteriori is used to classify the test sample.

No free lunch theorem refers to that there is no algorithm that has the best performance on any type of data. All learning algorithms have been better performance on particular data sets and have neither advantage on whole datasets. SVM-based classifiers are not proper for high-dimensional and big data sets. Decision trees are not able to detect complex decision boundary. Bias and variance dilemma should be done correctly in ensemble classifiers designing; noise and outlier detection is the most important problem in kNN classifier.

3 Proposed Method

The proposed method is based on CRF. First, we describe the CRF and Pairwise CRF in Sects. 3.1 and 3.2 describes the optimization problem for learning pairwise CRF parameters and solving it.

3.1 CRF and Pairwise CRF

The proposed method is based on Conditional Random Field. The CRF is a discriminative model for directly determines the posterior probability $P(Y|X)$ where Y is a set of target variables and X is a set of observed variables. More formally, a CRF is an undirected graph $G = (V, E)$ whose nodes correspond to $X \bigcup Y$; the network is annotated with a set of factors $\psi_1(T_1), \cdots, \psi_n(T_n)$ such that each $T_i \not\subseteq X$. The network represents a conditional distribution as follows:

$$\begin{aligned} P(Y|X) &= \frac{1}{Z(X)}\tilde{P}(Y,X) \\ \tilde{P}(Y,X) &= \prod_{i\in V}\psi_i(T_i) \\ Z(X) &= \sum_{Y}\tilde{P}(Y,X) \end{aligned} \tag{1}$$

mmeybodi@aut.ac.ir

where

– $Z(X)$ denotes the normalization factor

$$Z(X) = \sum_{y_1,\cdots,y_d} \prod_{i \in V} \psi_i(y_i, x) \tag{2}$$

– T represents the training dataset

$$T = \{(x^n; y_1^n, \cdots, y_d^n)\}_{n=1}^N \tag{3}$$

– and ψ_i is the node potential

$$\psi(y, x) = \exp(f_i(x)v_i^0, f_i(x)v_i^1) \tag{4}$$

where v_i^0, v_i^1 are the parameters of node i and $f_i(x)$ is the feature function.

Pairwise CRF is an extension of standard conditional random fields in which is also included the relationships between labels. More formally, a pairwise CRF defined as follows:

$$\begin{aligned} P(Y|X) &= \frac{1}{Z(X)} \tilde{P}(Y, X) \\ \tilde{P}(Y, X) &= \prod_{i \in V} \psi_i(T_i) \prod_{(i,j) \in E} \psi_{ij}(y_i, y_j, x) \\ Z(X) &= \sum_{Y} \tilde{P}(Y, X) \end{aligned} \tag{5}$$

The main difference between between standard CRF and pairwise CRF is the ψ_{ij} parameter which ψ_{ij} is the edge potential and calculated by the following equation:

$$\psi_{ij}(y_i, y_j, x) = \exp \begin{pmatrix} f_{ij}(x)e_{ij}^{0,0} & f_{ij}(x)e_{ij}^{0,1} \\ f_{ij}(x)e_{ij}^{1,0} & f_{ij}(x)e_{ij}^{1,1} \end{pmatrix} \tag{6}$$

where $(e_{ij}^{0,0}, e_{ij}^{0,1}, e_{ij}^{1,0}, e_{ij}^{1,1})$ are the parameters of edge i, j and f_{ij} is the feature function. There are several methods for parameter estimation which likelihood function is one of the conventional methods. This method is very computationally intensive and so extremely slow. One of the best approximation methods is the log pseudo likelihood function [3]. In next section describes the log pseudo likelihood (*lpl*) function and optimization problem for parameter estimation.

3.2 Parameter Estimation and Optimization Problem

Parameter estimation is the most crucial step in the CRF. As discussed in the previous section likelihood function is computationally expensive. For solving this problem we used *lpl* function as an approximation method.

mmeybodi@aut.ac.ir

Given i.i.d training data $\mathcal{T} = \{(x^n; y_1^n, \cdots, y_d^n)\}_{n=1}^N$, node and edge parameters can be estimated by:

$$lpl(T, \theta) = \sum_{n=1}^{N} \sum_{i=1}^{d} \log(P(y_i^n | y_{N(i)}^n, x^n)) \tag{7}$$

where $\theta = (v_i, e_{ij})$. Putting together 4 and 6, the $P(y_i^n | y_{N(i)}^n, x^n)$ can be written as:

$$P(y_i^v | y_{N(i)}^n, x^n) = \frac{exp(f_i(x^n) v_i^{y_i} + \sum_{j \in N_i} f_{ij}(x^n) e_{ij}^{y_i, y_j})}{Z_i} \tag{8}$$

Consequently, *lpl* can be written as:

$$\begin{aligned} lpl(T, \theta) = \sum_{i=1}^{d} \sum_{n=1}^{N} \Bigg(& f_i(x^n) v_i^{y_i^n} + \sum_{j \in N(i)} f_{ij}(x^n) e_{ij}^{y_i^n, y_j^n} \\ & - log \Bigg\{ exp(f_i(x^n) v_i^0 + \sum_{j \in N(i)} f_{ij}(x^n) e_{ij}^{0, y_j^n}) \\ & + exp(\sum_{j \in N(i)} f_{ij}(x^n) e_{ij}^{1, y_j^n} \Bigg\} \Bigg) \end{aligned} \tag{9}$$

As you can see *lpl* is stricktly convex (all local minimum are global minimum). In order to avoid overfitting we employ a penalized function. Hence, if $lpl(T, \theta)$ is the original objective function we optimize a penalized version $\hat{lpl}(T, \theta)$ instead, such that:

$$\hat{lpl}(T, \theta) = lpl(T, \theta) - P(\theta) \tag{10}$$

where

$$P(\theta) = \sum_{i=1}^{d} \lambda_v \|v_i\| + \lambda_e \sum_{j \in E} \|e_j\| \tag{11}$$

The tunning parameters λ_v and λ_e detemines the strength of penalty; lower values to less overfitting. For setting these parameters we used cross-validation. By defining $\tilde{lpl} = -\hat{lpl}$ and putting together 10 and 11, the optimization problem can be defined as:

$$\theta^* = argmin_\theta(\tilde{lpl}(T, \theta) + P(\theta)) \tag{12}$$

We use Frank-Wolfe algorithm for solving this optimization problem. The Frank-Wolfe algorithm is an iterative algorithm for convex optimization which doing the following steps for solving the optimization problem 12:

- **Step1**: Let $\theta^0 = \{\theta, v, e\}$ as a possible solution and stopping condition ϵ
- **Step2**: Compute the objective function 12
- **Step3**: Update θ^k
- **Step4**: If $\theta^k - \theta^{k-1} \geq \epsilon$ goto step 2; otherwise end

In this section we introduce the pairwise CRF model for MLC and PFP then parameterized this model and finally proposed the optimization problem to obtain the model parameters and solved it by Frank-wolfe algorithm.

mmeybodi@aut.ac.ir

4 Experimental Results

To evaluate the proposed method, we select three popular classifiers in multi-label data field such as Rank-SVM, AD-Tree and ML-KNN to compare with our proposed method. In AD-Tree method that uses decision trees for MLC, epochs are adjusted equaled with 50. The ML-KNN is to calculate the Euclidean distances between instances and number of the closest neighbors (instances) is $k = 10$.

4.1 Evaluation Criteria

To compare the results with other methods, several evaluation criteria for MLC can be found in [9]. Among these criteria, three criteria are important which include:

- Hamming Loss(HL): The Hamming loss function computes the average Hamming distance between two sets of samples. If $\hat{y}_j$ is the predicted value for the j^{th} label of a given sample, y_j is the corresponding true value, and d is the number of classes or labels, then the Hamming loss between two samples is defined as:

$$HL(y_j, \hat{y}_j) = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{xor(y_j, \hat{y}_j)}{L} \tag{13}$$

  where L is the number of labels.
- Ranking Loss(RL): The Ranking loss function computes which averages over the samples the number of label pairs that are incorrectly ordered, i.e. true labels have a lower score than false labels, weighted by the inverse number of false and true labels. Formally, given a binary indicator matrix of the ground truth labels $y \in \{0,1\}^{|T|*d}$ and the score associated with each label $\hat{f} \in R^{|T|*d}$, the ranking loss is defined as:

$$\begin{aligned} RL(y, \hat{(f)}) &= \frac{1}{|T|} \sum_{i=0}^{|T|-1} \frac{1}{|y_i|(d - |y_i|)} S_i \\ S_i &= \left| \left\{ (k,l) : \hat{f}_{ik} < \hat{f}_{il}, y_{ik} = 1, y_{il} = 0 \right\} \right| \end{aligned} \tag{14}$$

  where $|\cdot|$ is the ℓ_0 norm or the cardinality of the set.
- Average Precision(AP): This criteria corresponds to the area under the precision-recall curve. It is used here to measure the effectiveness of the label rankings.

4.2 Data Set and Assessment Results

Standard datasets used in other similar research were used here. The characteristics of the datasets are reported in Table 1. HL, AP and RL comparison between the proposed method and other methods is given in Tables 2, 3 and 4 respectively. The proposed method has been compared with four well-known methods.

mmeybodi@aut.ac.ir

It has been tried to select the methods which are applicaple. In AD-Tree method that uses decision trees for multi-label data classification, epochs are adjusted to be equaled with 50. The purpose of ML-KNN is to calculate the Euclid distances between instances. The number of closet neighbors (instances) is k=10. In Rank-SVM we use RBF Kernel or the equation $k(x, y) = exp(-\gamma \parallel x - y \parallel_2^2)$.

Table 1. Characteristics of different datasets

Dataset	#Instance	#Attribute	#Label
Cellcycle (fun facts)	3757	77	499
Cellcycle (go)	3751	77	4125
Derisi (fun facts)	3725	63	1275
Derisi (go)	3719	63	4119
Yeast	1484	103	14
Dorothea	1950	10000	51

HL criterion is percentage of labels which could not be predicted for proper instance. Table 2 presents the results associated to executing the proposed method and other methods for hamming loss criteria. The lower value of the size of criteria leads to the better efficiency in the algorithm. Evidently, the HL of the proposed method is acceptable with datasets Yeast, Dorothea, Derisi (fun facts) and Cell cycle (fun facts).

Table 2. Hamming loss

Dataset	ML-KNN	AD-tree	Rank-SVM	One.vs.Rest	Proposed Method
Cellcycle (fun facts)	0.233	0.285	0.243	0.310	0.228
Cellcycle (go)	0.359	0.471	0.301	0.367	0.324
Derisi (fun facts)	0.286	0.257	0.359	0.304	0.227
Derisi (go)	0.401	0.559	0.488	0.478	0.443
Yeast	0.231	0.350	0.199	0.273	0.171
Dorothea	0.159	0.133	0.121	0.161	0.089

AP is obtained by average fraction of labels ranked to get a specific relevant label, $k \in L_i$, which are actually grouped in L_i. In Table 3, the results of our proposed method and other methods are shown. The AP of the proposed method is acceptable with three data sets of six datasets.

RL determines how many pairs of relevant and irrelevant criteria of each instance take the higher rank of corresponding labels. If the value of the criteria goes to smaller sizes, better efficiency will be appeared in algorithm. In Table 4, the results of the proposed method and other methods are represented based on ranking loss criteria.

mmeybodi@aut.ac.ir

Table 3. Average precision

Dataset	ML-KNN	AD-tree	Rank-SVM	One.vs.Rest	Proposed Method
Cellcycle (fun facts)	0.621	0.681	0.750	0.639	0.686
Cellcycle (go)	0.522	0.409	0.599	0.497	0.550
Derisi (fun facts)	0.744	0.600	0.600	0.733	0.780
Derisi (go)	0.500	0.584	0.622	0.610	0.601
Yeast	0.770	0.723	0.740	0.644	0.731
Dorothea	0.830	0.761	0.839	0.890	0.915

Table 4. Ranking loss

Dataset	ML-KNN	AD-tree	Rank-SVM	One.vs.Rest	Proposed Method
Cellcycle (fun facts)	0.23	0.23	0.19	0.22	0.26
Cellcycle (go)	0.30	0.26	0.35	0.21	0.29
Derisi (fun facts)	0.24	0.29	0.18	0.30	0.24
Derisi (go)	0.30	0.22	0.24	0.38	0.19
Yeast	0.18	0.21	0.18	0.16	0.16
Dorothea	0.19	0.15	0.13	0.22	0.13

5 Conclusion

In this paper, we proposed a pairwise conditional random fields for protein function prediction. Based on this approach, we implemented a multilabel classifier for protein function prediction which considered the correlation among the labels. We successfully tested the performance of this classifier on three well-known classifiers. Given the positive results on Hamming loss, Average precision, and Ranking loss, we can conclude that proposed method is appropriate. We will recommend ways to improve the efficiency of MLC and PFP in the future. Scalable and efficient parameter estimation techniques for computing and feature learning can be employed to increase the performance.

References

1. Dessimoz, C., Škunca, N.: The Gene Ontology Handbook. Humana Press, New York (2017)
2. Moyano, J.M., Gibaja, E.L., Cios, K.J., Ventura, S.: Review of ensembles of multi-label classifiers: models, experimental study and prospects. Inf. Fusion **44**, 33–45 (2018)
3. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT press, Cambridge (2009)
4. Yu, G., Domeniconi, C., Rangwala, H., Zhang, G., Yu, Z.: Transductive multi-label ensemble classification for protein function prediction. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1077–1085. ACM (2012)

mmeybodi@aut.ac.ir

5. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: European Conference on Machine Learning, pp. 406–417. Springer (2007)
6. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural information processing systems, pp. 681–687 (2002)
7. Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 42–53. Springer (2001)
8. Zhang, M.-L., Zhou, Z.-H.: Ml-knn: a lazy learning approach to multi-label learning. Pattern Recogn. **40**(7), 2038–2048 (2007)
9. Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning. Pattern Recogn. **45**(9), 3084–3104 (2012)

mmeybodi@aut.ac.ir

Adversarial Samples for Improving Performance of Software Defect Prediction Models

Z. Eivazpour[1(✉)] and Mohammad Reza Keyvanpour[2]

[1] Department of Computer Engineering and Data Mining Laboratory, Alzahra University, Tehran, Iran
z.eivazpour@student.alzahra.ac.ir
[2] Department of Computer Engineering, Alzahra University, Tehran, Iran
keyvanpour@alzahra.ac.ir

Abstract. Software defect prediction (SDP) is a valuable tool since it can help to software quality assurance team through predicting defective code locations in the software testing phase for improving software reliability and saving budget. This leads to growth in the usage of machine learning techniques to SDP. However, the imbalanced class distribution within SDP datasets is a severe problem for conventional machine learning classifiers, since result in the models with poor performance. Over-sampling the minority class is one of the good solutions to overcome the class imbalance issue. In this paper, we propose a novel over-sampling method, which trained a generative adversarial nets (GANs) to generate synthesized data aimed for output mimicked minority class samples, which were then combined with training data into an increased training dataset. In the tests, we investigated ten freely accessible defect datasets from the PROMISE repository. We assessed the performance of our offered method by comparing it with standard over-sampling techniques including SMOTE, Random Over-sampling, ADASYN, and Borderline-SMOTE. Based on the test results, the proposed method provides better mean performance of SDP models among all tested techniques.

Keywords: Software defect prediction · Generative Adversarial Nets · Class imbalance · Over-sampling

1 Introduction

With the fast evolution in complexity and size of today's software, the prediction of defect-prone (DP) software artifacts play a crucial role in the software development process [1]. Current SDP work focuses on (i) Estimating the number of remaining defects, (ii) Discovering the associations of defect(s) and artifacts, and (iii) Classifying the defect-proneness of software artifacts, typically into two classes, DP and not defect-prone (NDP) [2]. In this paper, we concentrated on the third approach.

Classification approach of SDP task can help to the software developers and the project manager to prevent defects by suggesting that personnel focus more on these artifacts in order to find defects, efficiently prioritizing testing efforts and assign the

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 299–310, 2020.
https://doi.org/10.1007/978-3-030-37309-2_24
mmeybodi@aut.ac.ir

limited testing resources to them [3–5]. In the context of constructing the predictors, practitioners and researchers have applied numerous statistical and machine learning techniques (e.g., Neural Networks, Naïve Bayes, and Decision Trees) [6, 7]. Amongst them, the machine learning techniques are the most prevalent [1, 8], due to their efficiency. However, the vital problem in most of the standard learning techniques is that they tend to amplify the overall predictive accuracy. However, the accuracy of the classifiers is often obstructed through the imbalanced nature of the SDP datasets [9, 10]. Class imbalance is a state in which the data of some classes are much fewer than those of other classes [11]. In SDP, this issue is that DP class data are less than the NDP class ones [12–14]. Therefore, models trained on imbalanced datasets are ordinarily biased towards the NDP class samples and ignore the DP class samples [15], and it leads to the poor performance of SDP models [16, 17]. Thus, a good learner to be applied for SDP should provide high predictive accuracy of the minority samples (DP software artifacts), whereas conserving low predictive error rate of the majority samples (NDP software artifacts).

There are many studies to address the class imbalance learning in SDP. The prevalent approach to solving the problem of class imbalance is to use data sampling techniques; because of easy to use. The most popular among them being the over-sampling techniques, whereby new synthetic or artificial data samples are intelligently introduced into the minority (DP) class samples. These synthetic methods tend to introduce some bias towards the DP class, thus improving the performance of prediction models in the DP class.

An approach to the data generation would be the usage of a generative model that captures the original data distribution. Generative Adversarial Networks (GANs) [18] are composed of two networks, a discriminative one and a generative one, which competes against one another. Usually, the two adversaries are multilayer perceptrons. In this paper, to address the imbalanced dataset problem, we apply a GANs in creating synthesized data. This is the first attempt to usage GANs in SDP.

We conducted practical experiments to illustration the performance of the offered method in comparison to four common over-sampling approaches: Random Over-Sampling (ROS), SMOTE, Borderline-SMOTE (BSMOTE) and ADASYN using ten imbalanced datasets from the PROMISE repository[1] and considering two machine learning algorithms assessed on the resampled datasets. Our results show that our method improves the performance mean for all tested models.

The rest of the paper is structured as follows. Section 2 provides an overview of the existing over-sampling methods for SDP. Section 3 offers an overview of GANs. Our proposed method is described in Sect. 4. Section 5 provides an explanation of used datasets. Section 6 offers a description of reported evaluation measures. Section 7 presents the details of the experiments. Section 8 provides the results of the experiment, and Sect. 9 concludes the paper and summarize future work.

[1] http://openscience.us/repo/.

mmeybodi@aut.ac.ir

2 Related Work

There are various studies on applying Over-sampling techniques to SDP that we presented a summary of several studies in following.

Random Over-Sampling (ROS) randomly duplicates the minority data to increment the minority samples. However, ROS increases no new or further information to the classifier as the datasets consist of duplicates and consequently lead to over-fitting [19]. An improved method developed by Chawla et al. [20] called as Synthetic Minority Over-sampling TEchnique (SMOTE), augments the minority class data by producing new synthetic samples via considering vital information of the dataset. This technique generates new samples along a line segment that joins each sample and some specific k minority class nearest neighbors samples. Several variants of SMOTE followed for modifications. Han et al. [21] proposed the Borderline-SMOTE method, which creates synthetic samples along the line separating the data of two classes in a bid to strengthen the minority data found on the decision border. He et al. [22] proposed the Adaptive synthetic sampling approach (ADASYN) wherein used a weighted distribution method, which assigns weights associated with the learning characteristics of the minority class data. Bennin et al. [23] introduced MAHAKIL approach that uses features from two-parent samples to create a new synthetic sample based on their Mahalanobis distance and thus synthetic samples have the features of both parent samples. Rao et al. [24] offered ICOS (Improved Correlation over Sampling) approach that uses over-sampling strategy to produce new samples applying synthetic and hybrid category approaches. Huda et al. [25] applied different over-sampling techniques to create an ensemble classifier. Recently, Malhotra et al. [26] proposed the method SPIDER3 as modifications in SPIDER2 algorithm [27], as another attempt for the oversampling methods. Eivazpour et al. [28] proposed a new oversampling technique with applying generators to create synthesized samples in SDP field, that trained a Variational Autoencoder (VAE) aimed for output mimicked minority samples which were then united with the training set into an increased training samples set.

3 Overview of Generative Adversarial Nets

Generative Adversarial Networks (GANs) offer a method in order to learn a continuous-valued generative model sans a fixed parametric form to the output. This is done by means of establishing a generator function which maps from latent into data space and a discriminator function which maps from the data space toward a scalar. The discriminator function D attempts to predict the probability of the input data being from p_{data}, and the generator function G is trained to maximize the discriminator error on G(z). Both the discriminator and the generator are parametrized in the form of neural networks. The value function of the min-max game can be defined as follows:

$$\min_{G} \max_{D} V(D,G) = E_{x\sim P_{data}(x)}[log(D(x))] + E_{z\sim P_z(z)}[log(1-D(G(z)))]. \quad (1)$$

mmeybodi@aut.ac.ir

where D(x) characterizes the probability that x came from the original data distribution rather than the modeled distribution through the generator. In practice, at the start of generated samples from G are extremely poor and refused by D with high confidence rate. It has been observed to work fine in practice in order that the generator aimed at maximizing log(D(G(z))) quid pro quo minimizing log(1 − D(G(z))). In training (1) is resolved by alternating the subsequent two gradient update steps:

$$Step1: \theta_G^{t+1} = \theta_G^t - \lambda^t \nabla_{\theta_G} V(G^t, D^t), \tag{2}$$

$$Step2: \theta_D^{t+1} = \theta_D^t + \lambda^t \nabla_{\theta_D} V\left(G^{t+1}, D^t\right). \tag{3}$$

where θ_G and θ_D are the parameters of G and D, t is the iteration number, and λ is the learning rate.

Goodfellow et al. [18] demonstrate that, given enough capacity to G and D and sufficient training iterations, by a random vector, z, the network G can synthesize an example, which resembles one that is formed from the true distribution. Figure 1 shows the structure of GANs.

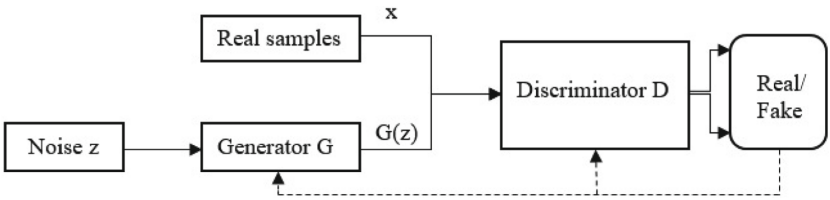


Fig. 1. An overview of the computation procedure and the structure of GANs [29].

4 Proposed Approach

To tackle the imbalanced problem, GANs can be used to generate synthetic samples for the minority class by receiving the z random noise vector. The D is set up as a binary classifier to distinguish fake and real minority class samples, and the G is set up in the role of an over-sampled data generator that can be difficulty predicted by D. The final generative model is applied to create synthetic data on the DP class as close as possible, and D, with similar network architecture, regards the samples as real data. We then combined synthetic samples with original training data, so that the desired effect can be attained by means of traditional classification algorithms. Our proposed approach can be depicted as Algorithm 1.

The main idea of existing over-sampling methods is that generating new samples be close in the aspect of "distance measurement" to available the DP class samples. The

mmeybodi@aut.ac.ir

different of existing methods, our proposed method is based on a latent probability distribution learned of data space, instead of being based on a pre-defined "distance measurement".

Algorithm 1 Overview of Proposed Approach(N, z)

Input: training set of minority and majority class samples N; random noise vector z
Output: Balanced training set
Procedure Begin
1) Split training set N into arrays of minority class N_{Def} and majority class N_{ND}
2) Counter ← Compute the number of additional minority class samples to be generated
3) Use N_{Def} set as the training set for a GAN, and tuning its hyperparameters
4) generate synthetic samples New_{Def} using the generator of the GAN with considering Counter
5) assign all samples in New_{Def} with the label of the minority class (defective) and Add them to the training set

End

5 Datasets Description

To simplify the verification and replication of investigates, the proposed method was examined on ten freely available benchmark datasets from PROMISE Repository were used. Details of them are given in Table 1. The first column includes the datasets names. The second column describes the number of features. The third column defines instances exists, and the latter two columns offer the number of DP instances and the percentage of DP instances distribution, respectively.

Table 1. Details of the datasets used

Dataset	# Features	# Instances	# Minority instances	%Defective
KC1	21	2109	326	15.5
KC2	21	522	107	20.5
KC3	39	194	36	18.5
MC1	38	9466	68	0.7
MC2	39	161	52	32.3
MW1	37	403	31	7.7
PC1	21	1109	77	6.9
PC2	36	5589	23	0.4
PC3	37	1563	160	12.4
PC4	37	1458	178	12.2

6 Model Evaluation Measures

To evaluate the performance usually is used the confusion matrix that displayed in Table 2.

mmeybodi@aut.ac.ir

Table 2. Confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

The SDP models effectiveness is assessed using measurements based on the confusion matrix; e.g., classifier accuracy, the number of predicted defects (Pd), and the number of erroneously predicted samples as no defects (Pf). Accuracy is the ratio of the properly predicted defects. In other words, it assesses the discriminating ability of the classifier. Pd is the ratio of properly predicted defects to the entire number of defects. Pf is the number of NDP artifacts that are classified erroneously as defects. The total Accuracy, Pd, and Pf are defined as Eqs. 4, 5, and 6, respectively.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$Pd = \frac{TP}{TP + FN} \tag{5}$$

$$Pf = \frac{FP}{FP + TN} \tag{6}$$

Since total accuracy, Pd, and Pf is not applicable for imbalanced datasets, we used Area Under the ROC Curve (AUC) measure [11, 30]. The AUC computed from the Receiver Operating Characteristics (ROC) curve. In other words, it is the trade-offs among the true and false positive error rates. The AUC is a value amid 0 and 1.

7 Experiments

We further preprocessed the datasets to remove duplications and used the z-score function to detect outlier samples and to scale the features into the interval [0, 1] using Eq. (7):

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{7}$$

where x is a feature and comprises (x_1,…,x_n), and max(x) and min(x) are the maximum and minimum values of the feature. To assess the performance of the models k-fold cross-validation strategy was applied with k = 10. To get reliable results, the experimental procedure was repeated 30 times, each time the instances ordering was shuffled, and the results average values between the experiments reported. The GANs implementation was

mmeybodi@aut.ac.ir

based on TensorFlow library [31]. The Python package "imbalanced-learn" [32] is used for implementations of the existing methods (ROS, SMOTE, BSMOTE, and ADASYN). "K" parameter in this implementation related to K-nearest neighbor algorithm set to 5. The used classifiers are set with default values of their parameters. We used the average presented by Python package "scikit-learn" [33] to calculate AUC values. Procedure 1 displayed experiments procedure.

The generator and discriminator are a 3-layer perceptron. Instability issue during the GANs training solved through the fine-tuning of the hyper-parameters. Each layer active function is ReLu [34]. Adam optimizer [35] is used for the optimizer. Initially, the weights of the networks are set randomly, the biases are adjusted to zero, and momentum was set to 0.5. The range for the number of epochs was found to be 500–4,000 and the learning rate is 0.03 were defined. Batch size is set into 42. Choosing the dimension of the noise vector z, the Number of units for a hidden layer of G and D. The resulting values are depicted in Table 3. Note that altogether values were initiated empirically.

Procedure 1: Experiments Process

```
M ← 30;   /*the number of iterations*/
N ← 10;   /*the number of folds in k-fold cross-validation strategy*/
D ← Original Datasets;
Classifiers ← Machine Learning Algorithms;
OversamplingTechniques ← The Used Over-sampling Methods;
for each dataset ϵ D do
    Remove repeated instances;
    Apply the z-score normalization to detect and delete outlier samples;
    Rescale dataset via using a min-max function;
    for each times ϵ [1, M] do
        bindataset ← separate N bins from dataset;
        for each fold ϵ [1, N] do
            trainingSet ← dataset - bindataset [fold];
            for each Classifier ϵ Classifiers do
                Predictor ← Classifier(trainingSet);
                ClassifierPerformance ← evaluate Predictor on bindataset [fold];
            end for
            for each imbMethod ϵ OversamplingTechniques do
                for each Classifier ϵ Classifiers do
                    SynthesizedData ← generateData(trainingSet, imbMethod);
                    Predict ← Classifier(SynthesizedData);
                    Performance ← evaluate Predict on bindataset [fold];
                end for
            end for
        end for
    end for
end for
```

mmeybodi@aut.ac.ir

Table 3. The dimension of noise vector z and the number of hidden units for G and D.

Dataset	The dimension of the noise vector	The number of hidden layer's units for G	The number of hidden layer's units for D
KC1	80	160	80
KC2	80	160	80
KC3	130	65	40
MC1	1200	1600	950
MC2	35	90	45
MW1	180	540	270
PC1	15	50	80
PC2	850	950	650
PC3	80	160	80
PC4	80	160	80

8 Results

In Tables 4 and 5, we exhibit the comparison results of the average AUC values of the proposed method with ROS, SMOTE, ADASYN, BSMOTE and No Sampling (NONE) by applying two machine learning algorithms, e.g., Decision Trees (DT) and Random Forest (RF).

Table 4. Average AUC values for DT classifier of the over-sampling techniques.

Dataset	None	ROS	SMOTE	BSMOTE	ADASYN	Our method
KC1	0.619	0.642	0.691	0.671	0.687	0.740
KC2	0.717	0.739	0.735	0.762	0.737	0.801
KC3	0.642	0.576	0.554	0.595	0.582	0.699
MC1	0.750	0.782	0.811	0.832	0.834	0.872
MC2	0.604	0.607	0.642	0.642	0.632	0.679
MW1	0.450	0.588	0.631	0.634	0.628	0.670
PC1	0.588	0.675	0.648	0.649	0.649	0.703
PC2	0.450	0.482	0.591	0.589	0.578	0.636
PC3	0.628	0.581	0.652	0.659	0.662	0.708
PC4	0.793	0.751	0.771	0.768	0.751	0.813

We found that the imbalanced datasets of SDP field yielded the poorest values, regardless of the used classifier. It can be understood which the AUC evaluation metric values of other methods are in general lesser in comparison to the proposed over-sample method when DT and RF are applied as a machine learning algorithm. This can be described by means of the closeness and uniqueness of the new synthetic samples produced by GANs in the defect dataset. Especially, RF has superior performance results with our proposed method (see Fig. 2). Note that the classifiers and datasets in

mmeybodi@aut.ac.ir

Table 5. Average AUC values for RF classifier of the over-sampling techniques.

Dataset	None	ROS	SMOTE	BSMOTE	ADASYN	Our method
KC1	0.664	0.679	0.699	0.699	0.705	0.754
KC2	0.718	0.716	0.732	0.741	0.736	0.768
KC3	0.535	0.617	0.645	0.597	0.588	0.677
MC1	0.871	0.883	0.902	0.913	0.918	0.949
MC2	0.686	0.690	0.731	0.729	0.721	0.768
MW1	0.730	0.702	0.736	0.733	0.732	0.771
PC1	0.669	0.676	0.689	0.646	0.650	0.724
PC2	0.607	0.644	0.761	0.746	0.759	0.793
PC3	0.783	0.810	0.830	0.828	0.834	0.872
PC4	0.812	0.810	0.816	0.821	0.824	0.844

the work [28] and this paper are the same, and comparing the results of both proposed methods indicates that the GANs generator performs better than the VAE generator in generating minority (DP) samples.

The misclassification happening in SDP field can be divided into two types of errors, known as "Type-I" and "Type-II", and consequently there are two kinds of misclassification costs. Type I cost is the misclassification cost of an NDP artifact as a DP artifact (e.g., false positive), whereas the type II cost is misclassification cost of labeling a DP artifact as an NDP artifact (e.g., false negative). Prior cause waste of resources related to testing. The latter origins we to lose the chance to modify the defect artifact before delivery to the customer and defects revealed through the customer are usually to fix expensive and damage the credibility of the software company. It is understandable that the costs related to the Type II misclassification are much higher

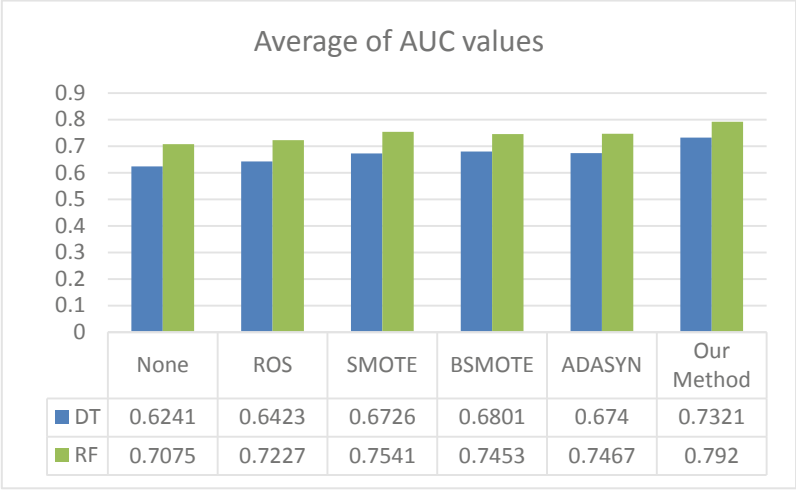


Fig. 2. Comparisons among DT and RF across five over-sampling methods.

mmeybodi@aut.ac.ir

than the costs related to the Type I, and therefore it is necessary that the Type II misclassification reduced to results in cost savings for a software development group. Our method minimizes the type II misclassification; since the related average of AUC values is more than other methods.

9 Conclusion and Future Work

In this study, we propose the use of GANs, as a state-of-the-art over-sampling method to address the issue of binary class-imbalanced data in SDP field. Given a training set, an augmented dataset is made, comprising more examples of the minority class samples regarding the original dataset. Synthetic samples are created by a GANs. The proposed method performance was assessed through comparing it to existing over-sampling techniques (ROS, SMOTE, ADASYN, and Borderline-SMOTE) over ten imbalanced datasets of SDP field, with applying Decision Tree and Random Forest as classifiers. The results prove that the proposed method outperforms in comparison with the other existing methods.

As the future work, we would like to fine-tune presented models and investigation about how to incorporate a weighted loss function into a model. We would like to test the ability of other classifiers to predict when they are learned on the artificial samples created through GANs and the verification of our proposed method on other defect datasets.

References

1. Zheng, J.: Predicting software reliability with neural network ensembles. Expert Syst. Appl. **36**, 2116–2122 (2009)
2. Song, Q., Jia, Z., Shepperd, M., Ying, S., Liu, J.: A general software defect-proneness prediction framework. IEEE Trans. Softw. Eng. **37**(3), 356–370 (2011)
3. Abaei, G., Selamat, A.: A survey on software fault detection based on different prediction approaches. Vietnam J. Comput. Sci. **1**, 79–95 (2014)
4. Clark, B., Zubrow, D.: How good is the software: a review of defect prediction techniques. Sponsored by the US Department of Defense (2001) 12
5. Wang, S., Liu, T., Tan, L.: Automatically learning semantic features for defect prediction. In: Proceedings of the 38th International Conference on Software Engineering, pp. 297–308. ACM (2016)
6. Khoshgoftaar, T.M., Allen, E.B., Deng, J.: Using regression trees to classify fault-prone software modules. IEEE Trans. Reliab. **51**(4), 455–462 (2002)
7. Porter, A.A., Selby, R.W.: Empirically guided software development using metric-based classification trees. IEEE Softw. **7**(2), 46–54 (1990)
8. Menzies, T., Greenwald, J., Frank, A.: Data mining static code attributes to learn defect predictors. IEEE Trans. Softw. Eng. **33**(1), 2–13 (2007)
9. Gray, D., Bowes, D., Davey, N., Sun, Y., Christianson, B.: The misuse of the NASA metrics data program data sets for automated software defect prediction. IET Semin. Dig. **1**, 96–103 (2011)

mmeybodi@aut.ac.ir

10. Bennin, K.E., Keung, J., Monden, A., Kamei, Y., Ubayashi, N.: Investigating the effects of balanced training and testing datasets on effort-aware fault prediction models. In: Proceedings of the 40th Annual Computer Software and Applications Conference, vol. 1, pp. 154–163. IEEE (2016)
11. He, H., Garcia, E.: Learning from imbalanced data. IEEE Trans. Data Knowl. Eng. **21**(9), 1263–1284 (2009)
12. Shuo, W., Xin, Y.: Using class imbalance learning for software defect prediction. IEEE Trans. Reliab. **62**(2), 434–443 (2013)
13. Sun, Z., Song, Q., Zhu, X.: Using coding-based ensemble learning to improve software defect prediction. J. IEEE Trans. Syst. Man Cybern. Part C **42**, 1806–1817 (2012)
14. Fenton, N.E., Ohlsson, N.: Quantitative analysis of faults and failures in a complex software system. IEEE Trans. Softw. Eng. **26**(8), 797–814 (2000)
15. Provost, F.: Machine learning from imbalanced data sets 101. In: Proceedings of the AAAI 2000 Workshop on Imbalanced Data Sets, pp. 1–3 (2000)
16. Hall, T., Beecham, S., Bowes, D., Gray, D., Counsell, S.: A systematic literature review on fault prediction performance in software engineering. IEEE TSE **38**(6), 1276–1304 (2012)
17. Arisholma, E., Briand, L.C., Johannessen, E.B.: A systematic and comprehensive investigation of methods to build and evaluate fault prediction models. J. Syst. Softw. **83** (1), 2–17 (2010)
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS, pp. 2672–2680 (2014)
19. García, V., Sánchez, J., Mollineda, R.: On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. Knowl. Based Syst. **25**(1), 13–21 (2012)
20. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
21. Han, H., Wang, W., Mao, B.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005)
22. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the International Joint Conference on Neural Networks, 2008, Part of the IEEE World Congress on Computational Intelligence, Hong Kong, China, 1–6 June 2008, pp. 1322–1328 (2008)
23. Bennin, K.E., Keung, J., Phannachitta, P., Monden, A., Mensah, S.: Mahakil: diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction. IEEE Trans. Softw. Eng. **44**(6), 534–550 (2018)
24. Rao, K.N., Reddy, C.S.: An efficient software defect analysis using correlation-based oversampling. Arabian J. Sci. Eng. **43**, 4391–4411 (2018)
25. Huda, S., Liu, K., Abdelrazek, M., Ibrahim, A., Alyahya, S., Al-Dossari, H., Ahmad, S.: An ensemble oversampling model for class imbalance problem in software defect prediction. IEEE Access **6**, 24184–24195 (2018)
26. Malhotra, R., Kamal, S.: An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. Neurocomputing **343**, 120–140 (2019)
27. Napierała, K., Stefanowski, J., Wilk, S.: Learning from imbalanced data in presence of noisy and borderline examples. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 158–167. Springer, Heidelberg (2010)
28. Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., Wang, F.: Generative adversarial networks: introduction and outlook. IEEE/CAA J. Automatica Sinica **4**, 588–598 (2017)

mmeybodi@aut.ac.ir

29. Eivazpour, Z., Keyvanpour, M.R.: Improving performance in software defect prediction using variational autoencoder. In: Proceedings of the 5th Conference on Knowledge Based Engineering and Innovation (KBEI), pp. 644–649. IEEE (2019)
30. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst. Man Cybern. Part C **42**(4), 463–484 (2012)
31. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., Brain, G.: TensorFlow: a system for large-scale machine learning. In: OSDI, pp. 265–284 (2016)
32. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. **18**(17), 1–5 (2017)
33. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
34. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: AISTATS 2011, pp. 315–323 (2011)
35. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

mmeybodi@aut.ac.ir

A Systematic Literature Review on Blockchain-Based Solutions for IoT Security

Ala Ekramifard, Haleh Amintoosi(✉), and Amin Hosseini Seno

Computer Engineering Department, Faculty of Engineering,
Ferdowsi University of Mashhad, Mashhad, Iran
{ekramifard,amintoosi,Hosseini}@um.ac.ir

Abstract. Nowadays, we are facing an exponential growth in the Internet of Things (IoT). There are over 8 billion IoT devices such as physical devices, vehicles and home appliances around the world. Despite its impact and benefits, IoT is still vulnerable against privacy and security threats, due to the limited resources of most IoT devices, lack of privacy considerations and scalability issues, thus making traditional security and privacy approaches ineffective for IoT. The main goal of this article is to investigate whether the Blockchain technology can be employed to address security challenges of IoT. At first, a Systematic Literature Review was conducted on Blockchain with the aim of gathering knowledge on the state-of-the-art usages of Blockchain technology. We found 44 use cases of Blockchain in the literature from which, 18 use cases were specifically designed for the application of Blockchain in addressing IoT-related security issues. We classified the state-of-the-art use cases in four domains, namely, smart cities, smart home, smart economy and smart health. We highlight the achievements, present the methodologies and lessons learnt, and identify limitations and research challenges.

Keywords: Internet of things · Blockchain · Security · Privacy

1 Introduction

Internet of Things represents a collection of heterogeneous devices that communicate with each other automatically. It has been widely used in many aspects of human life, such as industrial, healthcare systems, environmental monitoring, smart city, building and smart home, etc. IoT devises create, collect, and process privacy-sensitive information and send them to cloud via Internet, generating mass of valuable information that can be targeted by attackers. Moreover, most IoT devices have limited power and capacity, making the usage of traditional security solutions computationally expensive. In addition, most well-known security solutions are centralized and are not compatible with the distributed nature of IoT. Hence, there is a vital demand for a lightweight, distributed and scalable solution to provide IoT security.

Blockchain is a distributed ledger that is used to provides electronic transactions security, and at the same time guaranteeing the auditability and nonrepudiation. In Blockchain, cryptography is used to provide secure ledger management for each node without needing a central manager. Blockchain can help to develop decentralized

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 311–321, 2020.
https://doi.org/10.1007/978-3-030-37309-2_25
mmeybodi@aut.ac.ir

applications running on billions of devices. It has prominent features such as security, immutability and privacy, and therefore can be a useful technology to address the security challenges of many applications.

In this paper, we performed a systematic literature review on the state-of-the-art to investigate the possibility of leveraging Blockchain to provide security and privacy for IoT applications in four categories: smart cities, smart home, smart economy and smart health.

The rest of the paper is organized as follows. The main structure of IoT and Blockchain as well as the research goal and research questions of the systematic review and its process are expressed in Sect. 2. Section 3 presents an overview on the Blockchain-based security solutions for IoT, according to the above-mentioned categories and the results obtained from the literature review. Section 4 presents the conclusion and open challenges.

2 Research Design

2.1 IoT Security and Privacy

IoT contains heterogeneous devices with embedded sensors interconnected through a network, which are uniquely identifiable and mostly characterized by low power, small memory and limited processing capability. The gateways are deployed to connect these devices to the cloud for remote provision of data and services to users [1].

IoT applications have very different objectives, from a simple appliance for a smart home to equipment for an industrial plant. Generally, IoT operations include three distinct phases: collection phase, transmission phase, and processing and utilization phase. Sensing devices, which are usually small and resource constrained, collect data from environment. Technologies for this phase operate at limited data rates and short distances, with constrained memory capacity and low energy consumption. These collected data transmit to applications with transmission technologies that are more powerful. In last phase applications process collected data to obtain useful information and take decisions to controlling the physical objects and act on the environment [2].

Due to development of hardware and network facilities, the use of IoT is expanding rapidly in everyday life. Hence, providing security and privacy in this field is very important. Security and privacy are fundamental principles of any information system. Security is the combination of integrity, availability, and confidentiality that can be obtained by authentication, authorization, and identification. Privacy is defined as the right that an individual has to share his information [3].

There are three main challenges in IoT that make traditional security solutions ineffective. First, most IoT devices have limited bandwidth, memory and computation capability which makes them inefficient for complex cryptographic algorithms. Second, IoT is subject to scalability challenge since there may be billions of devices connecting to a cloud server that may result in bottleneck problem. Third, devices normally report raw data to the server, resulting in the violation of users' privacy. Therefore, new security technologies will be required to protect IoT devices and platforms. To ensure the confidentiality, integrity, and privacy of data, proper encryption mechanisms are

mmeybodi@aut.ac.ir

needed. To secure communication between devices and privileged access to services, the authentication is required. Various mechanisms need to guarantee availability of services and prevent denial-of-service, sinkhole, replay, and others attacks. Various components of IoT like applications, framework, network, and physical devices have specific vulnerabilities and variety of different solutions have been implemented. A comprehensive review on security issues in IoT has been presented in [1].

2.2 Blockchain

Blockchain is a decentralized, distributed, and immutable database ledger that stores transactions and events in a peer-to-peer network. It is known as the fifth evolution of computing, the missing trust layer for the Internet. Bitcoin was the first innovation that introduced Blockchain. It is a decentralized cryptocurrency, which can be used to buy and exchange goods [3].

Blockchain is chained blocks of stored data transactions that are validated by miners. Each block includes a hash, time stamped sets of recent valid transactions, and the hash of the previous block. When a user requests a transaction, first it is transmitted to the network. The network checks it for validation and the valid transaction is added to the current block and then chained to the older blocks of transactions [4].

Blockchain provide immutability and verifiability by mixing hash functions and Merkle trees. Hash is the one-way mapping function, which transforms data of any size into short, fixed-length values. Merkle tree takes many hashes and squeezes them to one hash. To construct a new Merkle tree, leaf nodes that contain data are hashed and parent nodes combine pairs of hashes to calculate a new hash node. This process is continued until the root of the tree is constructed. Each block in Blockchain contains the root of this tree as well as all transactions within the block [4, 5].

Blockchain can be built as a private network that can be restricted to a certain group of participants, or public network that is open for anyone to join in like Bitcoin [1]. Blockchain does not have a central authority. In public Blockchain, when participants are anonymous, a malicious attacker may want to corrupt the history of data. Bitcoin for example, prevents this by using a consensus mechanism called proof of work (PoW) which is the Byzantine problem solving. Every machine that stores a copy of the ledger tries to solve a complex puzzle based on its version of the ledger. The first machine who solves the puzzle wins and all other machines update their ledgers with winner [4].

Blockchain has some advantages over existing electronic frameworks like transparency, low or no exchange costs, network security and financial data assurance [3]. In addition to cryptocurrencies applications, public ledger and a decentralized environment can be used in various applications like IoT, smart contracts, smart property, and digital content distribution [6]. When information has been written into a Blockchain database, it is nearly impossible to remove or change, so it leads to trust in digital data. Therefore, data is reliable and we can transact business online.

mmeybodi@aut.ac.ir

2.3 Research Goal and Questions

The goal of this research is to investigate whether and to what extend the Blockchain technology is able to address the security and privacy issues of IoT. At first, a Systematic Literature Review was conducted on Blockchain with the aim of gathering knowledge on the state-of-the-art usages of Blockchain technology. To do so, we considered the following research questions:

- **RQ1:** How does Blockchain address the security and privacy issues of IoT in the domain of smart home?
- **RQ2:** How does Blockchain address the security and privacy issues of IoT in the domain of smart city?
- **RQ3:** How does Blockchain address the security and privacy issues of IoT in the domain of smart health?
- **RQ4:** How does Blockchain address the security and privacy issues of IoT in the domain of smart economy?

2.4 Search Process

The searches were conducted on 25th October 2018 and we processed all studies that had been published up to this date. To obtain the collection of relevant studies, we selected the key terms "Blockchain" and "IoT" to search in IEEE Xplore, ScienceDirect, DBLP, and Google Scholar. The searches were run against the title, keywords and abstract. Duplicate studies were removed. The result was around 600 papers.

The results from these searches were filtered through the inclusion/exclusion criteria.

Inclusion Criteria are: (1) The paper must be an empirical study about usage of Blockchain in IoT applications. (2) The paper must emphasize on security. (3) The paper must have enough details about its approach.

Exclusion Criteria are: (1) Conference version of a study that has an extended journal version. (2) Non-English language papers.

The 'Abstract' of all 600 papers were read. After running the studies through the inclusion/exclusion criteria, 44 papers were remained for reading. These 44 papers were fully read and inclusion/exclusion criteria were re-applied, leaving 18 papers that specifically address the security issues related to the usage of Blockchain in IoT applications. It is worth mentioning that we conducted snowballing to make sure that no further papers were detected that met the inclusion criteria. (To be brief, snowballing refers to using the reference list of a paper or the citations to the paper to identify additional papers.) (Table 1).

mmeybodi@aut.ac.ir

Table 1. Use cases of Blockchain in IoT security

Paper	Category	Usage of blockchain
[8]	Smart City	Increase security and privacy in vehicular ecosystem
[9]	Smart City	Distributed transport management
[10]	Smart City	Secure data transfer in Internet of Vehicles
[7, 14]	Smart City	Smart Energy Grid
[11, 12]	Smart City	Trusted data sharing environment
[15–17]	Smart Home	Secure lightweight architecture
[18]	Smart Home	Self-management identity
[19]	Smart Home	Authentication and secure communication
[20]	Smart Health	Control and share health data in an easy and secure way
[21]	Smart Health	Securely share health data
[22, 23]	Smart Health	Access control
[28]	Smart Economy	Reliable energy trading
[29]	Smart Economy	Anonymous energy trading

3 Review Results

In this section, we review the related articles related to each research question and discuss the results.

3.1 RQ1: Use Cases Related to Smart City

Smart City is aimed at improving the life quality of citizens by integrating the ICT services and urban infrastructures. There are a large number of smart city applications and technologies to realize complex interactions between citizens, third parties, and city departments and most of them heavily rely on data collection, interconnectivity, and pervasiveness. Despite the benefits, they are major threats to the privacy of citizens [7].

Common security and privacy methods that are used in smart city tend to be ineffective due to some challenges, such as single point of failure centralized communication model and lack of users' privacy. Hence, decentralized privacy preserving and secure Blockchain-based architecture can help to face these challenges.

Authors in [8–10] proposed architectures based on Blockchain technology with the aim of preserving the security and privacy of vehicular systems. The paper in [8] discusses the efficacy of the proposed security architecture in some smart city applications like wireless remote software updates. To provide integrity, all transactions contain the hash of the data. Similarly, to provide confidentiality, transactions are encrypted via utilizing asymmetric encryption.

Authors in [9] proposed a reliable and secure vehicle network architecture based on Blockchain to build the distributed transport management system. In this model, to achieve scalability and high availability of the vehicle network, there are three kinds of nodes: controllers that are connected in a distributed manner to provide the necessary services on a large scale, miner node, which handles request/response requests and vehicle nodes that are just ordinary nodes which send a service request message either

mmeybodi@aut.ac.ir

to miner or controller nodes. Controllers process and compute the data (including a hash, a timestamp, a nonce, and a Merkle root) and share it to other nodes in a distributed manner. All communications are encrypted using the public/private keys to secure the privacy of the client's data.

Communication between vehicles must be secure to prevent malicious attacks, and it can be achieved by authenticating all nodes before connecting to the network. An authentication and secure data transfer algorithm, was proposed in Internet of Vehicles using the Blockchain technology in [10]. Each vehicle is made to register with the Register Authority (RA) to prevent any malicious vehicle to become a part of the network.

Authors in [11] proposed a data-sharing environment for intelligent vehicles that is aimed to provide the trust environment between the vehicles based on Blockchain. To ensuring secure communication between vehicles, this mechanism provides ubiquitous data access based on crypto unique ID and an immutable database. They also proposed Intelligent Vehicle Trust Point (IV-TP) mechanism, which provides trustworthiness for vehicles behavior [12]. IV-TP is an encrypted unique number, which is generated by the authorized authority. To provide secure vehicles communication, it uses Blockchain as follows: each vehicle generates its private and public key, and then digitally signs messages to ensure integrity and non-repudiation. Receiver verifies the digitally signed message and decrypts it.

Authors in [13] introduced a Blockchain-based intelligent transportation system, which is a seven-layer conceptual model. It consists of a physical layer that encapsulates data of various kinds of physical entities such as devices and vehicles. The data layer produces chained data blocks by using asymmetric encryption, time-stamping, hash algorithms and Merkle tree techniques. The network layer is responsible for communication among entities, data forwarding and verification. Consensus Layer includes various consensus algorithms like PoW and PoS. Incentive layer includes issuance and allocation mechanisms of economic reward of Blockchain. Contract Layer controls and manages physical and digital assets. Application Layer includes application scenarios and use cases.

The article in [14] used Blockchain to recharge the autonomous electric vehicles in intelligent transportation systems. This system includes three parts: a particular charging station as server, vehicles as client, and a smart contract. Charging station and cars communicate with each other through the channel that is opened and prices are per unit of charging. Other parameters have been set in a Blockchain as contract.

A Smart Energy Grid technology was proposed in [7] to improve the energy distribution capability for citizens in urban areas. The proposed method uses the Blockchain technology to join the Grid, exchange information, and buy/sell energy between energy providers and private citizens. *From review the literature in the domain of smart city, we conclude that the Blockchain can improve security in smart city specifically in two ways: secure data transfer in vehicular ecosystem and autonomous electric charging. Moreover, via Blockchain, the need for centralized companies to entrust users' data is eliminated.*

mmeybodi@aut.ac.ir

3.2 RQ2: Use Cases Related to Smart Home

Smart home is equipped with a number of IoT devices including a smart thermostat, smart bulbs, an IP camera and several other sensors. Smart devices should be able to store data on storages to be used by a service provider. Collection, processing and dissemination of data may result in the revealing of private behavior and lifestyle patterns of people [15].

Several works have addressed the challenges in ensuring security and privacy for smart home. A secure lightweight Blockchain-based architecture for smart home has been proposed in [15–17] that eliminates the concept of POW and the need for coins, to decrease the overhead of Blockchain. This architecture consists of three main tiers namely: smart home, overlay network, and cloud storage. Each smart home is equipped with a high resource device called "miner" that is responsible for handling all communication within and external to the home [16]. Nodes in the overlay network are grouped in clusters, to decrease network overhead and delay. Devices can store their data in the cloud storage, so that a service provider can access this data and provide certain smart services [15]. This work mostly focuses on data store and access control, in IoT devices. Data storage and access transactions have been stored as transactions in the Blockchain. The public keys are fixed with the cluster head. The proposed model has been analyzed against DDOS and linking attacks and the overhead of using their model over traditional message exchange models has been measured.

To overcome the problems of centralized identity management systems which are built basis on third-party identity providers, authors in [18] have proposed a Blockchain-based Identity Framework for IoT (BIFIT) in smart home. It provides self-management identity for devices in IoT environment, and helps casual users without technical expertise to manage and control them. This framework includes an autonomic monitoring system that relies on digital signature to control appliance behavior in order to detect any suspicious activities. In addition, it develops a unique identity for each device to correlate with its owner for the sake of ownership and security management.

The paper in [19] proposed an authentication and secure communication scheme in smart home based on Extended Merkle Tree and Keyless Signature Infrastructure (KSI). It provides authentication with a public key-secret key structure and generates integrity of the message by KSI's distributed server using the global timestamp value. It improves efficiency by eliminating the structure of the existing PKI based certificate system. *To conclude: Blockchain can be used for secure authentication, access control and communication in the domain of smart home. The main challenge, however, is the scalability issue due to the large size of Blockchain and cryptographic solutions which is not suitable for IoT devices with limited resources.*

3.3 RQ3: Use Cases Related to Smart Health

Sharing healthcare information makes healthcare systems smarter and improves the quality of their services. The analysis and storage of healthcare data must be done in a secure way and should be kept private from other parties, as it may be used maliciously by attackers. To overcome these challenges, a Blockchain-based Healthcare Data Gateway (HDG) storage platform was proposed in [20] to enable patients to own,

mmeybodi@aut.ac.ir

control and share their own data in an easy and secure way without violating privacy. It consists of three layers. The Storage layer stores data in the private Blockchain cloud and protects data with cryptographic techniques thus ensuring the medical data cannot be altered by anybody. The data management layer works as a gateway and evaluates all data accesses. The data usage layer includes entities that use patient healthcare data.

Authors in [21] propose a secure healthcare system that is aimed at sharing health-related between the nodes in a secure manner. It contains two main security protocols: an authentication protocol between medical sensors and mobile devices in a wireless body area network and a Blockchain-based method to share heath data.

The work in [22] proposed a decentralized electronic medical records (MedRec) management system that was aimed handling secure information while managing security goals such as authentication, confidentiality and data sharing. It uses Ethereum as smart contract and stores information about ownership, permissions and integrity of medical records. It also uses cryptographic hash of the data to prevent tampering.

A secure, scalable access control mechanism for sensitive information has been proposed in [23]. It is a Blockchain-based data sharing method that permits data owners to access medical data from a shared repository after their identities and cryptographic keys have been verified. This system consists of three entities: users that want to access or contribute data, system management composed of entities responsible for identification, authentication and authorization process, and cloud-based data storage.

A softwarized infrastructure for secure and privacy preserving deployment of smart healthcare applications was proposed in [24]. The privacy of sensitive patient data is ensured using Tor and Blockchain, where Tor removes mapping between user IP address and Blockchain tracks and authorizes access to confidential medical records. This prevents records from being lost, wrongly modified, falsified or accessed without authorization. *To conclude: the most important security challenges in Smart Health are privacy preserving health data sharing, authorized access to such data and preserving the integrity of health data, From reviewing the literature in the domain of smart health, it has been documented that Blockchain-based solutions are able to guarantee the security requirements of health data to a great extent, without the need to trust a third party.*

3.4 RQ4: Use Cases Related to Smart Economy

Blockchain has been widely applied for financial transactions, generally called cryptocurrency. However, it is not the only use of Blockchain in economy. Researchers are trying to identify new solutions in various economic aspects utilizing Blockchain benefits. In fact, integrating IoT and Blockchain may lead to excellent opportunities to develop distributed shared economy. Automatic payment mechanisms, foreign exchange platforms, and digital rights management are some of these applications [25]. Blockchain can also be used to digitally track the ownership of assets across business collaborations or to capture information about the product from participants across the supply chain in secure and immutable manner [26].

Smart contract is a computerized transaction protocol that is written by users to be uploaded and executed on the Blockchain, so to increase the need for trusted intermediaries between parties [1]. Authors in [27] describe the benefit of Blockchain, IoT

mmeybodi@aut.ac.ir

and smart contract combination in automation of multi-step processes and marketing services between devices. ADEPT, Filament, Watson IoT platform, and IOTA are some other economic scenarios that are explained in [3]. ADEPT builds a network of distributed devices that transmit transactions to each other and perform maintenance automatically by use of smart contracts to provide security. Filament allows devices to interact autonomously with each other, for example to sell environmental conditions data to a forecasting agency. Watson platform provides a private Blockchain to push IoT devices data so that business partners can access them in a secure manner. IOTA is a cryptocurrency for selling the data that is collected from IoT devices [3].

A decentralized energy-trading platform, without reliance on trusted third party was implemented in [28]. It is a token-based private system and all trading transactions are done anonymously, and data is replicated among all active nodes to protect from failure. It uses Blockchain, multi-signatures and anonymity of users to provide privacy and security.

Using IoT devices as smart meters in Smart Grid can lead to energy trading without the need of third party. Authors in [29] have proposed a reliable infrastructure to transactive energy, based on Blockchain and smart contracts, which helps energy consumers and producers to sell to each other directly without the involvement of other stakeholders. Using Blockchain in this architecture leads to the increased reliability, higher cost effectiveness, and improved security. *To conclude: by utilizing the Blockchain applications such as cryptocurrency and smart contract, it is possible to improve the reliability of smart economy and add anonymous trading to economic systems.*

4 Conclusion

In this paper, we conducted a systematic literature review on the recent works related to the application of Blockchain technology in providing IoT security and privacy. The goal of our research is to verify whether the Blockchain technology can be employed to address security challenges of IoT. We selected 18 use cases that are specifically related to applying Blockchain to preserve IoT security and categorized them into four domains: smart home, smart city, smart economy and smart health. Due to the decentralized nature of Blockchain, its inherent anonymity afforded and the provided secure network on untrusted parties, it has been gaining great attention in addressing the security challenges of IoT. In fact, Blockchain technology facilitates implementation of decentralized Inter-net of things' platforms and allows secure recording and exchanging information. In this structure, the Blockchain plays the role of the ledger, and all exchanges of data on the intelligent devices are recorded safely. However, despite all the benefits, the Blockchain technology is not without shortcomings. Encryption that is used in Blockchain-based techniques is time and power consuming. IoT devices have very different computing capabilities, and not all of them are capable to run the encryption algorithms at the appropriate speed. Since Blockchain has a decentralized nature, scalability is one of the major challenges in this area. Size of the ledger will increase over time, and usually this size of data is more than the capacity of most IoT nodes. Since there are many nodes in IoT scenarios, we need a large number

mmeybodi@aut.ac.ir

of keys for secure transactions between devices. These issues introduce new research challenges. Moreover, with the increasing use of IoT devices in real world, the number of malicious attacks to these tools increases. Therefore, there is a need for extensive researches on vulnerabilities in current technologies and the identification and counteraction to attacks. Most recent works that rely on Blockchain just introduce models or prototypes, without dealing with real implementations. There seems to be a need for more research to examine the performance of new models and designs.

Conflict of Interest. On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Khan, M.A., Salah, K.: IoT security: review, blockchain solutions, and open challenges. Future Gener. Comput. Syst. **82**, 395–411 (2017)
2. Zarpelão, B.B., et al.: A survey of intrusion detection in internet of things. J. Netw. Comput. Appl. **84**, 25–37 (2017)
3. Jesus, E.F., Chicarino, V.R.L., de Albuquerque, C.V.N., Rocha, A.A.D.A.: A survey of how to use blockchain to secure internet of things and the stalker attack. Secur. Commun. Netw. **2018**, article ID 9675050, 27 p. (2018). https://doi.org/10.1155/2018/9675050
4. Laurence, T.: Blockchain for Dummies. Wiley, Hoboken (2017)
5. Chitchyan, R., Murkin, J.: Review of blockchain technology and its expectations: case of the energy sector. arXiv preprint arXiv:1803.03567 (2018)
6. Yli-Huumo, J., Ko, D., Choi, S., Park, S., Smolander, K.: Where is current research on blockchain technology?—a systematic review. PloS One **11**(10), e0163477 (2016)
7. Pieroni, A., et al.: Smarter city: smart energy grid based on blockchain technology. Int. J. Adv. Sci. Eng. Inf. Technol. **8**(1), 298–306 (2018)
8. Dorri, A., Steger, M., Kanhere, S.S., Jurdak, R.: Blockchain: a distributed solution to automotive security and privacy. IEEE Commun. Mag. **55**(12), 119–125 (2017)
9. Sharma, P.K., et al.: A distributed blockchain based vehicular network architecture in smart city. J. Inf. Process. Syst. **13**(1), 84 (2017)
10. Arora, A., Yadav, S.K.: Block chain based security mechanism for internet of vehicles (IoV). In: 3rd International Conference on Internet of Things and Connected Technologies, pp. 267–272 (2018)
11. Singh, M., Kim, S.: Blockchain based intelligent vehicle data sharing framework. arXiv preprint arXiv:1708.09721 (2017)
12. Singh, M., Kim, S.: Intelligent vehicle-trust point: reward based intelligent vehicle communication using blockchain. arXiv preprint arXiv:1707.07442 (2017)
13. Yuan, Y., Wang, F.Y.: Towards blockchain-based intelligent transportation systems. In: Intelligent Transportation Systems (ITSC), pp. 2663–2668 (2016)
14. Pedrosa, A.R., Pau, G.: ChargeltUp: on blockchain-based technologies for autonomous vehicles. In: The 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, pp. 87–92 (2018)
15. Dorri, A., Kanhere, S.S., Jurdak, R.: Blockchain in internet of things: challenges and solutions. arXiv preprint arXiv:1608.05187 (2016)
16. Dorri, A., et al.: Blockchain for IoT security and privacy: the case study of a smart home. In: IEEE Percom Workshop on Security Privacy and Trust in the Internet of Thing (2017)

mmeybodi@aut.ac.ir

17. Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P.: LSB: a lightweight scalable blockchain for IoT security and privacy. arXiv preprint arXiv:1712.02969 (2017)
18. Zhu, X., et al.: Autonomic identity framework for the internet of things. In: International Conference of Cloud and Autonomic Computing (ICCAC), pp. 69–79 (2017)
19. Ra, G.J., Lee, I.Y.: A study on KSI-based authentication management and communication for secure smart home environments. KSII Trans. Internet Inf. Syst. **12**(2) (2018)
20. Yue, X., Wang, H., Jin, D., Li, M., Jiang, W.: Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. J. Med. Syst. **40**(10), 218 (2016)
21. Zhang, J., Xue, N., Huang, X.: A secure system for pervasive social network-based healthcare. IEEE Access **4**, 9239–9250 (2016)
22. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A.: Medrec: using blockchain for medical data access and permission management. In: 2nd International Conference on Open and Big Data, IEEE, pp. 22–24 (2016)
23. Xia, Q., Sifah, E.B., Smahi, A., Amofa, S., Zhang, X.: BBDS: blockchain-based data sharing for electronic medical records in cloud environments. Information **8**(2), 44 (2017)
24. Salahuddin, M.A., Al-Fuqaha, A., Guizani, M., Shuaib, K., Sallabi, F.: Softwarization of internet of things infrastructure for secure and smart healthcare. arXiv preprint arXiv:1805.11011 (2018)
25. Huckle, S., Bhattacharya, R., White, M., Beloff, N.: Internet of things, blockchain and shared economy applications. Procedia Comput. Sci. **98**, 461–466 (2016)
26. How Blockchain Will Accelerate Business Performance and Power the Smart Economy (2017). https://hbr.org/sponsored/2017/10/how-blockchain-will-accelerate-business-performance-and-power-the-smart-economy. Accessed June 2018
27. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. IEEE Access **4**, 2292–2303 (2016)
28. Aitzhan, N.Z., Svetinovic, D.: Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans. Dependable Secure Comput. (2016)
29. Lombardi, F., Aniello, L., De Angelis, S., Margheri, A., Sassone, V.: A blockchain-based infrastructure for reliable and cost-effective IoT-aided smart grids. Living in the Internet of Things: Cybersecurity of the IoT (2018). https://doi.org/10.1049/cp.2018.0042

mmeybodi@aut.ac.ir



An Intelligent Safety System for Human-Centered Semi-autonomous Vehicles

Hadi Abdi Khojasteh[1], Alireza Abbas Alipour[1], Ebrahim Ansari[1,2(✉)], and Parvin Razzaghi[1,3]

[1] Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran
{hkhojasteh,alr.alipour,ansari,p.razzaghi}@iasbs.ac.ir
[2] Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Charles University, Prague, Czechia
[3] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
https://iasbs.ac.ir/~ansari/faraz

Abstract. Nowadays, automobile manufacturers make efforts to develop ways to make cars fully safe. Monitoring driver's actions by computer vision techniques to detect driving mistakes in real-time and then planning for autonomous driving to avoid vehicle collisions is one of the most important issues that has been investigated in the machine vision and Intelligent Transportation Systems (ITS). The main goal of this study is to prevent accidents caused by fatigue, drowsiness, and driver distraction. To avoid these incidents, this paper proposes an integrated safety system that continuously monitors the driver's attention and vehicle surroundings, and finally decides whether the actual steering control status is safe or not. For this purpose, we equipped an ordinary car called FARAZ with a vision system consisting of four mounted cameras along with a universal car tool for communicating with surrounding factory-installed sensors and other car systems, and sending commands to actuators. The proposed system leverages a scene understanding pipeline using deep convolutional encoder-decoder networks and a driver state detection pipeline. We have been identifying and assessing domestic capabilities for the development of technologies specifically of the ordinary vehicles in order to manufacture smart cars and eke providing an intelligent system to increase safety and to assist the driver in various conditions/situations. A pre-published version of this paper is available at arXiv website https://arxiv.org/pdf/1812.03953.pdf.

Keywords: Semi-autonomous vehicles · Intelligent Transportation Systems · Computer vision · Automotive safety systems · Self-driving cars

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 322–336, 2020.
https://doi.org/10.1007/978-3-030-37309-2_26

mmeybodi@aut.ac.ir

1 Introduction

According to the World Health Organization (WHO) in 2013, some 1.4 million people lose their lives in traffic accidents each year [26]. Also, a 2009 report published by the WHO had estimated that more than 1.2 million people die and up to 50 million people are injured or disabled in road traffic crashes around the world every year [27]. The statistics show that, due to the ever-increasing number of vehicles and density of traffic on roads, current intelligent transportation systems have been successful. However, the systems need to be further developed to decrease the number and severity of road accidents.

The Integrated Vehicle Safety System (IVSS) [11] is used for safety applications in vehicles. The system which includes various safety systems such as anti-lock braking system (ABS), emergency brake assist (EBS), traction control system (known as ASR), crash mitigation systems, and lane keeping assist systems. The purpose of an IVSS is to provide all safety related functions for all types of vehicles at a minimum cost. Such system offers several advantages like low cost, compact size, driving comfort, traffic information, and safety alerts. It also indicates the health of the car electrical components and provides information regarding an overall condition of the vehicle.

In the past decade, many studies have examined the advantages of integrated safety and driver acceptance along with integrated crash warning systems.



Fig. 1. The instrumented vehicle and drone (top right) with the vision system consists of four mounted cameras and a drone camera along with a universal car tool for communicating and sending commands to the vehicle. The front (top left) and rear (bottom left) wide-angle HD cameras are mounted at close to the center of the windshields. The driver-facing camera (bottom left) is mounted on the center of the roadway view. The car cabin camera (bottom right) is mounted on the center of the headliner to include a view of the driver's body.

mmeybodi@aut.ac.ir

A study was undertaken by the U.S. Department of Transportation indicates that number of crashes can be reduced significantly by developing collision warning systems to alert drivers of potential rear-end, lateral drift, and lane-changing crashes [11]. Such integrated warning systems will provide comprehensive and coordinated information, which can be used by crash warning subsystems to warn the driver of the foreseeable threats. For an intelligent system to reduce accidents and casualties, at least two general trends can be expected: *(1) Autonomous cars, (2) Driver assistance systems.*

An autonomous car also known as a self-driving car is a vehicle that has the characteristics of a traditional car and in addition, is capable of transporting automatically without human intervention. A driverless car system drives the vehicle by the perception of the environment and based on dynamic processes which result in steering and navigating the car to safety [4,12]. As it seems, the studies done to get to self-driving cars have led to creating the driver assistance systems. From another perspective, an utter vehicle control system, without examining different driver assistance systems, as well as the use of intelligent highways, is meaningless. Design and implementation of automated driving in the real environment, with regards to today's technology is still in the preliminary stages [6] and there is a long road to its full implementation. As a short-term and practical solution, today, much effort is being made in the research and industrial communities to design and implement driver assistance systems. For example, the automation of some areas of vehicle control, such as auto-steering, or the movement direction, has been widely studied and implemented or studies carried out on various driving maneuvers such as overtaking and automatic parking of the vehicle [18].

Deep learning [14,19,25] can be used to analyze and process input data received from the automobile sensors such as cameras, motion sensors, laser light, GPS, Odometry, LiDAR, and radar sensors and control the vehicle in response to information from various sensors [4,10,12,13,28]. Also, we can utilize computer vision for eye-gaze tracking [10,13,25,28], monitoring threshold blinking [22] and head movement [24] which in turn such in-car sensing technologies would enable us to warn the driver of drowsiness or distraction in real-time. Hence, these measures can be highly effective in avoiding collisions and reducing fatal accidents.

Our central goal in this work is to create a semi-autonomous car by integrating some state-of-the-art approaches in computer vision and machine learning for assisting the drivers during critical and risky moments in which driver would be unable to steer the vehicle safely. All supplementary materials are available for public access on the web[1].

The rest of the paper is organized as follows. We review former state-of-the-art approaches in Sect. 2. In Sect. 3, we describe our safety system architecture on the semi-autonomous car in which we applied some subtle manipulations along with ordinary capabilities of traditional vehicles. Section 4 explains our fine-grained system in detail. The paper concludes with Sect. 5 where we discuss the outcome and possible work to be done in the future.

[1] https://iasbs.ac.ir/~ansari/faraz.

mmeybodi@aut.ac.ir

2 Background

There are many works on preventing car accidents some of which deal with the effects of driver behavior in traffic accidents. As in [16] authors have carried out research on the drive in which they use the raw data that is collected for processing to define driving violations as a criterion for driving behavior and have examined the impact of various factors such as speed, the effect of density, velocity, and traffic flow on accidents. Much research has introduced automotive safety systems which designed to avoid or reduce the severity of the collision. In such collision mitigating systems, tools like radar, laser (LiDAR) and camera (employing image recognition) are utilized to detect an imminent crash [7]. Many articles have been presented to prevent crashes with the use of intelligent systems. Some systems react to imminent crash (occurring at the moment). As an example, in [6], using parameters like speed and distance of vehicles, the systems help prevent collisions at intersections or reduce damage and casualties. Some consider the current condition of the road and neighboring cars and using the available data, examine the probability of an accident and predict them to provide solutions to avoid accidents.

Moreover, an early work proposed a traffic-aware cruise control system for road transports that automatically tunes the vehicle speed to keep an assured distance from other cars ahead. Such systems might utilize various sensors such as radar, LiDAR, or a stereo camera system for the vehicle to brake when the system finds the vehicle is approaching another car ahead, then accelerate when traffic allows it to. One of the most common types of accidents is rear-end crashes which accounts for a significant percentage of accidents in different countries [18]. Rate of these accidents are even more frequent on the roads. In order to avoid rear-end accidents two solutions are considered: timely change of speed which is when the vehicle detects that a collision with the front (rear) vehicle is imminent, the speed is reduced/increased to prevent it, and changing the direction in order to prevent collisions with the front or rear car, the driver changes the car's path.

Most of the research focused on vision-based methods which used to assist the driver for steering a vehicle safely and comfortably. In [9] authors proposed an approach in which they use only cameras and machine learning techniques to perform the driving scene perception, motion planning, driver sensing to implement the seven principles that they described in the work for making a human-centered autonomous vehicle system [9]. Also author in [1] fused radar and camera data to improve the perception of the vehicle's surrounding, including road features and obstacles and pedestrians. As in [1,3,5,7,14,19] authors presented an assist system in which they utilize machine vision techniques to recognize road lanes and signs. These progressive image processing methods infer lane data from forward-facing cameras mounted at the front of the vehicle [1,3,7]. Some of the advanced lane finding algorithms have been developed using deep learning and neural network approaches [5,14,19]. Some other procedures used for monitoring the consciousness and emotional status of the driver are momentous for the safety and comfort of driving. Nowadays, real-time non-obtrusive

mmeybodi@aut.ac.ir

monitoring systems have been developed, which explore the driver's emotional states by considering facial expressions of them [9,10,13,21,24,28].

Given the nature of the safety and the fact that in previous studies the efficiency of presented methods for diagnosing the safety of car travels has been observed, hence we propose an integrated vehicle safety system, which is a compilation of the aforementioned approaches. This system is able to prove beneficial in terms of increasing the safety factor and driving safety and in turn, reducing crashes, casualties and the damage caused by accidents.

3 Architecture

A system we have built is composed of a variety of subsystems, which utilize the capabilities of the machine vision and factory-installed sensors information. The following, we describe the parts and implementation stages of the system in details.

3.1 Driving Scene Perception

As we steer a vehicle, we are deciding where to go by using our eyes. The road lanes are indicated by lines on the road, which work as stable references for where to drive the car. Intuitively, one of the first things we need to do in developing a self-driving car is to identify road lane-lines using an efficient algorithm. Here is a robust approach for driving scene perception that uses trained segmentation neural network for recognizing driving safe area and extracting road along with a lane detection algorithm to deal with the curvature of the road lanes, worn lane markings, emerging/ending lane-lines, merging, splitting lanes, and lane changes.

To identify lane-lines in a video that is recorded during car driving on the road, we need a machine vision method that performs detection and annotation tasks on every frame of the video in order to generate an appropriate annotated video. The method has a processing pipeline scheme that encompasses preliminary tasks like camera calibration and perspective measurement and later stages such as distortion correction, gradient, perspective transform, processing semantic segmentation output of the deep network and lane-line detection.

The lane-line finding and localizing algorithm must be effective for real-time detecting and tracking, and has an efficient performance for different atmospheric conditions, light conditions, road curvatures, and also for other vehicles, which are in road traffic. Here, we propose an approach relied on advanced machine vision techniques to distinguish road lanes from dash-mounted camera video and detect obstacles in the car's surroundings from both of front- and rear-camera. We utilize advanced computer vision methods to compute the curvature of the road, identify lanes, and also locate the vehicle in safe driving zone. At a glance. We pursued this process into three stages, in the first stage, we calibrate a front, rear, and top cameras with correct distortion of each frame of input video and create a more suitable image for subsequent processing. In the next stage, we

mmeybodi@aut.ac.ir

use a Deep Convolutional Encoder-Decoder Network that has an architecture inspired by ENet [20] and SegNet [2] to determine potential locations of the lane-lines in the image from full input resolution feature maps for pixel-wise classification. In the third stage we synthesize the lane mask information with prior frame information for computing the final lane-lines and identify the main vehicle's route, free-space and lane direction. This stage would be done to discard noisy effects, apply a perspective transform on the image, and track assigned lane and path (as shown in Fig. 2).

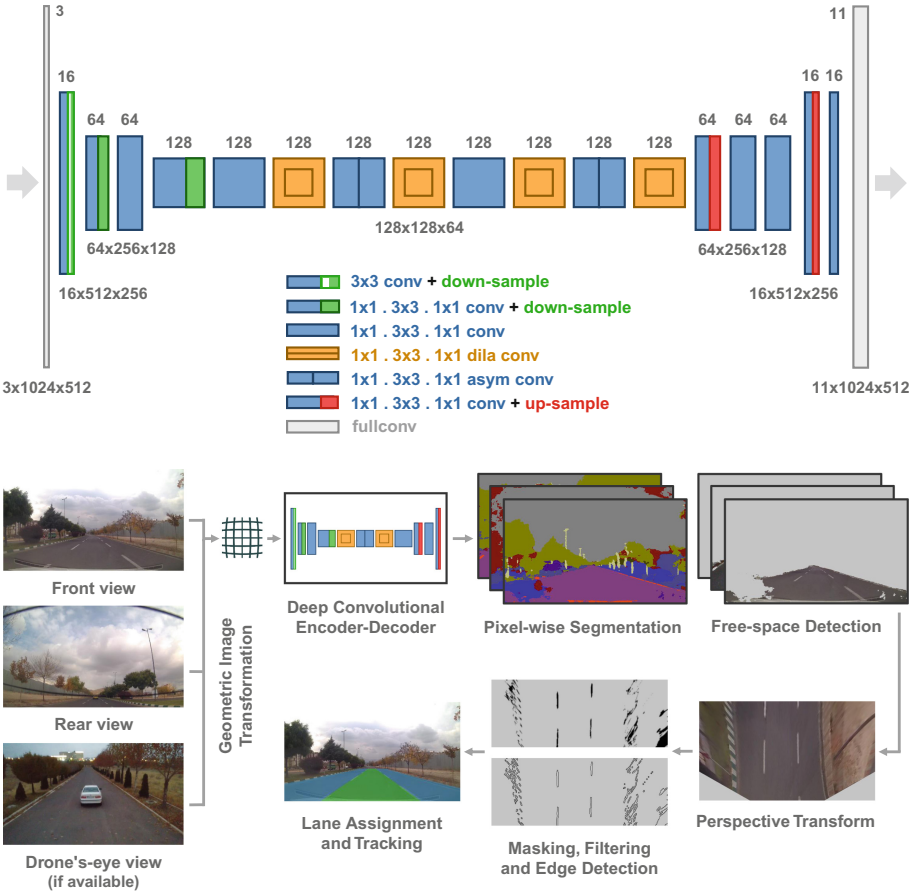


Fig. 2. The overall scene understanding pipeline along with architecture of the Convolutional Encoder-Decoder Network model for scene segmentation is shown in terms of layers of convolutional networks. Each block shows different types of convolution operations (normal, full, dilated, and asymmetric). The pipeline includes geometric transformation, encoder-decoder network, free-space detection, perspective transform, masking, filtering, edge detection, lane assignment, and tracking respectively.

mmeybodi@aut.ac.ir

For this pipeline, what steps are needed to do to get a better scene understanding that is to say: first, a new frame of the video is read and then undistorted by using precomputed camera distortion matrices based on our camera's intrinsic, and extrinsic parameters, which is known as undistort image.

At second stage, we propose a deep neural network with basic encoder-decoder architecture computational unit, consisting of 17 layers, and one dimensional convolutions with small convolutional operations. Hence, training and testing are accelerated and facilitated because of lower dimensional and small convolution operations. This model leverages various types of convolution operations that are consist of regular, asymmetric, and dilated. This diversity lessens the computational load by changing dimensions of 5×5 convolutions in a layer into two layers with 5×1 and 1×5 convolutions [23] and leads to fine-tuning the receptive field by the dilated convolutions application. The architecture of the encoder is similar to vanilla CNN, which includes several convolution layers with max-pooling. The encoder layers carry out feature extraction and pixel-wise classification of the down-sampled image. Somewhere else, the layers of the decoder do up-sampling after each convolutional layer for offsetting the encoder down-sampling effects and making an output with a size as same as the input. The beginning layer implements subsampling to diminish the computational load. The architecture as shown in Fig. 2 consists of 10 convolutional layers alongside max-pooling for the encoder, 5 convolutional layers in parallel with up-sampling belong to the decoder, and a conclusive 1×1 convolutional layer to combine the penultimate layer outputs. All the convolution operations are either 3×3 or 5×5, whereas 5×5 convolutions are asymmetric, that is to say, they are performed separately as 5×1 and 1×5 convolutions to lessen the computational load. Besides, some layers use dilated convolution to increment the effective receptive field of the associated layer. Therefore, this helps with growing faster the encoder receptive field without using down-sampling. Such model is highly efficient insomuch as all convolutions are either 3×3 or by 5×5 and collateral, in contrast to sequential, integration with max-pooling potentially retains inherent details of the environmental features.

The last stage is to compute lanes. Different lane calculations would be implemented for the first frame and subsequent frames. In the initial of this stage, we apply the perspective transform in which has given bird's eye view of the road that makes to discard any irrelevant information about the background from the warped image. In the next step, once we provide the perspective transform, next, we put on color masks to recognize yellow and white pixels in the image. For final step, besides the color masks, for detecting edges we apply some filters. We use the filters on L and S channels of the image since the filters made robust the color and lighting variations. Then, we merge candidate lane pixels from color masks, filters, and pixel-wise classification map to get potential lane regions. In the first frame, the lanes are computed and determined by computer vision methods. But, in the later frames, we tracked the location of the lane-lines from previous frame. This approach significantly reduces the computation time of the algorithm. Next, we introduced additional steps to ensure some errors which

mmeybodi@aut.ac.ir

might be occurred due to incorrectly detected lanes that would be removed. Last, the coefficients of the polynomial fit are used to compute curvatures of the lanes and relative location of the vehicle on the road lanes.

Ultimately, we gather all of the output results for three stages to determine the vehicle position on the road and detect free-space around the car for having a subtle defensive driving system.

3.2 Driver State Detection

Due to having a safe smart car, we should monitor the driver's behavior. An important component of the driver's behavior corresponds to eye-gaze tracking. Intuitively, the driver's allocation of visual attention away from the road is the momentous cause in increasing the hazards of driving. We are able to determine the status of the driver with their eye-gaze tracking and blink rate for detecting drowsiness and/or distraction. For monocular gaze estimation, we generally do the pupils locating and determine the inner and outer eye corners in driver's head image. Therefore, the eye corners would be as important as pupils and likely detecting them is more difficult rather than pupils. We describe how to extract the eye corners, eye region, and head pose and then utilize to estimate the gaze. The eye-gaze can be estimated using a geometric head model. If an estimate of the head pose is available, a more refined geometric model can be used and a more accurate gaze estimate is obtained. Of the locations of pupils, inner eye corner, outer eye corner, and head pose, the estimation of the eye corners is harder than other. Once the eye corners have been located, then, locating the pupils is done easily.

In recent years, there have been done a lot of work in face identification. A novel method is proposed in [15] shows that face alignment can be solved with a cascade of learnt regression functions, which be able to localize the facial landmarks when initialized with the mean face pose. In the algorithm, each regression function in the cascade meticulously assesses the shape from an initial approximation and the intensities of a sparse set of pixels indexed relevant to the initial assessment. We trained our face detector as a same approach in [15] by using a training set that is based on iBUG 300-W dataset, which used to learn the cascade. We determine the head pose by leveraging the proposed algorithm that estimated in a similar manner. At first, the algorithm detects and tracks a collection of anatomical feature points such as eye corners, nose, pupils, and mouth and then utilizes a geometric model to compute the head pose (as illustrated in Fig. 3).

The steps in the driver state detection pipeline are: face detection, face alignment, 3D reconstruction, and fatigue/distraction detection. First, step for face detector we use a Histogram of Oriented Gradients (HOG) along with a SVM classifier. In this step, a false detect can be costly in the single face and multiple faces case. For the single face case, the error leads to an incorrect gaze region prediction. In the multiple faces case, the video frame would be decreasing on consideration, which reduces true decision rate at the system. Then, we perform face alignment on a 56-point subset from the 68-point Multi-PIE facial landmark

mmeybodi@aut.ac.ir

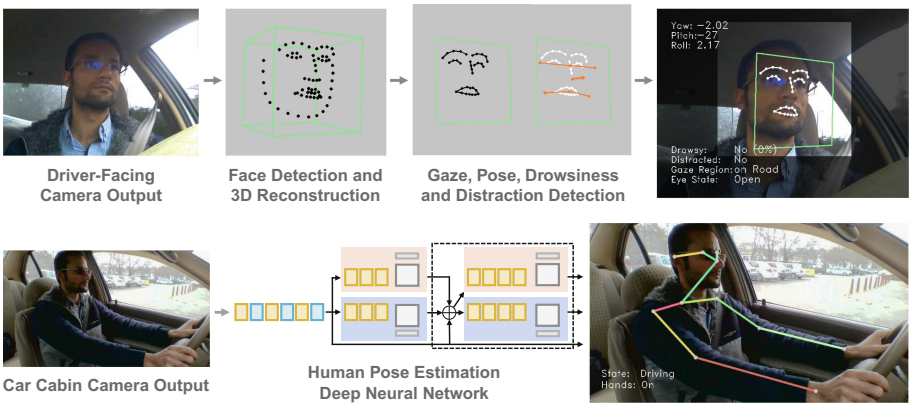


Fig. 3. Driver gaze, head pose, drowsiness and distraction detection implemented in real-time for low-illumination example (top row). The computed yaw, pitch, and roll are displayed on the top left and details of the predicted state are illustrated on the bottom left. The real-time model for driver body-foot keypoints estimation on car cabin camera RGB output (bottom row), which is represent by human skeleton including head, wrist, elbow, and shoulder by color lines.

markup used in the dataset. These landmarks include parts of the nose, upper edge of the eyebrows, outer and inner lips, jawline, and exclude all parts in and around the eye. Next, they would be mapped to a 3D model of the head. The resulting 3D-2D points correspondence can be used to compute the orientation of the head. This is categorized under geometric methods in [17]. The yaw, pitch, and roll of the head can be used as features for a gaze region estimation. By using these steps, our system is able to indicate a gaze region recognition for each image fed into the pipeline. Given fact that the driver spends more than 90% of their time looking forward at the road. We used this fact for normalizing facial features spot to the face bounding box, which corresponds to the road gaze region. In this step, we do not need calibration and just normalize the facial features based on eyes and nose bounding boxes only for the running frame. Eyes and nose bounding boxes are empirically found to be the most robust normalizing region. We should consider the fact that the big disorderliness in the face alignment step is correlated with the features of the jawline, the eyebrows, and the mouth.

The detected points are used to recognize eye closes and blinks. According to head pose in 3D space, we are able to track eye-gaze and diagnose either the driver is looking forward to the road or not. Thus, we will be able to indicate fatigue or distraction. Also, we leverage a deep neural network to perform a driver pose estimation for detecting the position and 3D orientation from major parts/joints of the body-foot keypoints (i.e. wrist, elbow, and shoulder), which

mmeybodi@aut.ac.ir

is represented by human skeleton. By using this model, we are able to identify the status of the driver's hands and how it is positioned.

In this section, we characterized how we are able to utilize these algorithms to make a gaze assessment system that derives a desirable precision from the fact that we would be localized the corners of the eye and head pose by using the face entire appearance, rather than by just exploring a few solitary points of them.

3.3 In-Vehicle Communication Device

One of the main ability of an active safety system is reliability and real-time communicating with the vehicle. In order to achieve more safety in driving with existing vehicles, we need to robust communicating with the vehicle system. For this reason, Universal Vehicle Diagnostic Tool (known as UDIAG) is developed as shown in Fig. 4, it is able to communicate with several types of vehicle internal communication network protocols. UDIAG connects to vehicle system via OBD-II standard connector of the vehicle directly, also it is able to connect to other types of connector via an external interface (Fig. 5), and negotiates with Electronic Control Units (ECUs) of the in-vehicle network according to own database. UDIAG translates data of the network into the useful and pure information such as parameter and fault codes of the vehicle and sends information via WiFi to other parts of safety system. Also, this platform injects command of safety system into in-vehicle network and saves a log of the network on own storage.

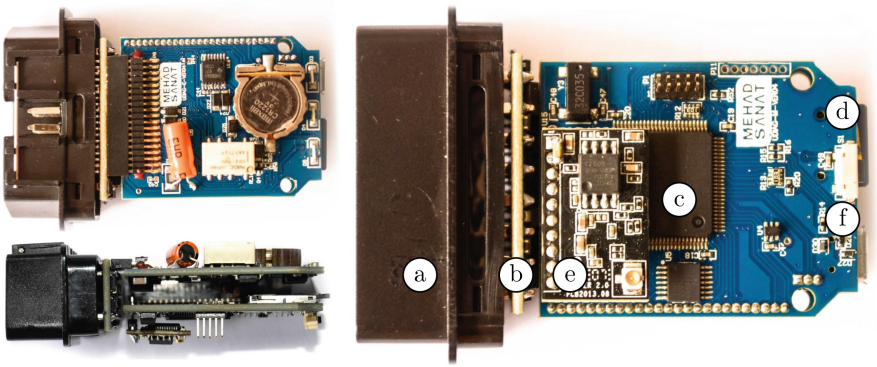


Fig. 4. The top, bottom, and left view of the Universal Vehicle Diagnostic Tool (known as UDIAG) that connects to vehicle diagnostic port and establishes communications with the in-vehicle network. The vehicle network interface (a), power supply (b), processing unit (c), data storage (d), wireless adapter (e) and Micro USB socket (f) are shown in the figure.

mmeybodi@aut.ac.ir



Fig. 5. UDIAG external interfaces with other types of connector instead of OBD-II standard connector for communicating with various vehicles.

UDIAG consists of five main parts: power supply, processor, in-vehicle network interface, storage, and a wireless interface (shown in Fig. 4). The power supply can support both 12v and 24v vehicles. UDIAG has an ARM cortex M4 (STMF407VGT) processor, the vehicle network interface supports KWP2000, ISO 9141, J1850 and CAN [8] physical layer. For storage it uses high-speed microSD card and utilizes WiFi-UART Bridge and USB for communicating with other parts of the safety system.

We leverage the UDIAG to receive information and gather data from vehicle control units, car systems and surrounding sensors along with mounted cameras, and then process and integrate them by our system, which ultimately leads to the issuance of appropriate commands (e.g. alerting the driver to drowsiness or sudden lane changes) in various conditions.

4 Implementation Details

In order to get a safety auto-steering vehicle without the use of specific and complex infrastructures, we need to design a system that has a thorough perception of the environment and car surroundings (i.e. the road, pedestrians, other vehicles, and obstacles) at least as much as a safety threshold. Therefore in our system implementation due to take an affordable project completion, we used only passive sensors, cameras, factory-installed in-vehicle sensors, low-cost device, and an ordinary laptop in the vehicle, which allow our proposed system to be easily implemented and exploited with low operational costs. The architecture of the system which is installed and tested on the FARAZ vehicle is based on an Intel Core i5 processor along with four cameras which are consisting of two wide-angle high-definition (HD) cameras, a night vision camera and webcam, and also a Universal Vehicle Diagnostic Tool called UDIAG. Two HD cameras are mounted at close to the top center of the windshield and rear window that are used for taking videos from front perspective to detect the road and lane-lines and rear perspective to detect other vehicles and obstacles in car's surroundings. One camera is mounted on dashboard to supervise the face of the driver for detecting fatigue, drowsiness and/or driver's distraction. Another webcam is mounted on the headliner close to the top center of the windshield that used for driver body pose monitoring. C++ programming language, OpenCV (Open

mmeybodi@aut.ac.ir

Source Computer Vision Library), and FreeRTOS (Free Real Time Operating System) have been used for a complete implementation of the system.

The process of the system is such that all data are collected from sensors and commands are received from the user interface, which can be entered through the system's control panel namely graphical user interface or keyboard. After analyzing input data, the system leverages extracted information to decide on the measures to perform regarding suited warnings and driving strategies. Also, for debugging purposes, a visual output would be supplied to the user and intermediate results are logged. FARAZ, shown in Fig. 1 is an experimental semi-autonomous vehicle equipped with a vision system and a supervised steering capability. It is able to determine its position with respect to road lane-lines, compute the road geometry, detect generic obstacles on the trajectory, and assign the vehicle to a lane and maimtain the optimal path. The system is designed as a safety enhancement unit. In particular, it is able to supervise the driver behavior and issue both optic and acoustic warnings. It issues a proper command/alert at speeding, sudden road lane changes, encountering an obstacle on car's route, when approaching to a vehicle's rear or vice versa and the possibility of rear-end collision, sudden crash around the car, when to drive slower than traffic, and even the need to fix the automobile using the information, which is acquired from car systems.

We are able to adjust the system to steer the car in two different modes: a *manual mode* that the system monitors and logs the driver's activity, and alerts of hazard cases to driver with acoustic and optical warnings. Data logging while driving in the system includes important data such as speed, lane detection and changes, user interventions, and commands. *Semi-automated mode* that in addition to warning and log capabilities, it also sends some controlling commands to car systems and even is able to take control of the vehicle when a dangerous situation is detected and also we equipped the car FARAZ with emergency devices that can be activated manually in case of system failures. Further, for future work, we will add an automated mode in the system that leads to full control on the vehicle.

The FARAZ car being used in our tests has eight ECUs for various tasks: Central Communication Node (CCN) in the dashboard to manage the central locking and alarm system and to communicate with the body modules, the lighting system, and read the status of the various switches, Door Control Node (DCN) for controlling door actuators and vehicle mirrors, Front Node (FN) on front of the vehicle to control the alternator, cooler compressor, horn and lights set, car alarms, and front actuators, Instrument Cluster Node (ICN) to control various front-end amps, Rear Node (RN) in the rear luggage compartment for rear-end car sensors and lights, Anti-lock Braking System (ABS) for the management of brakes and vehicle wheels, Airbag Control Unit (ACU) for airbags and related actuators, and Engine Management System (EMS), which is responsible for driving the engine vehicle and sending control commands. The status information and values of the actuators and car sensors associated with these

mmeybodi@aut.ac.ir

modules are read from the internal vehicle network and sent to the integrated safety system for decision making.

Values or status of vehicle speed, engine speed, engine status, throttle position, throttle angle, acceleration pedal angle, battery voltage, mileage, gearbox ratio and engine configuration from EMS, the speed of each wheel individually from ABS, the relevant information for each airbag from ACU, information on all switches (e.g. the wash pump, wiper, air condition, screen heater) inside the vehicle and in the car bonnet, information on the status of all car lamps (such as main, dipped, fog, side, hazard), hand break and brake pedal status, shock sensor status, seat belt status, gasoline level, the status of each car doors and mirrors, outdoor and indoor temperature, brake oil level, oil pressure, cruise control and target velocity of cruise control (if available) from CCN, DCN, FN, RN, and ICN nodes, and also the status of the central locking (Locked/Unlocked) and key position are obtained from the Immobilizer indirectly. Our device is also able to send appropriate commands for each of the actuators associated with each of the different modules according to the decision-making conditions.

Our vehicle has had decentralized road tests within a month in Zanjan. Each part of the system, as described in the previous section, was tested on the training data and validated before the final test on the vehicle. Then all of them have been put together to check the functionality of the system. The initial tests were conducted to check the overall performance of the vehicle along with a driver at all times of the test, on the campus paths and urban roads in a controlled environment of possible incidents (pedestrian crossings, car accidents, etc.). These tests were carried out at different times of the day and night with a cover distance of 100 km in normal climate conditions. In the future, these tests would be carried out on a long-term schedule. It also seeks to further implement this system on a commercial vehicle with more ECUs and more environmental sensors to add fully autonomous system capabilities.

5 Conclusions and Future Work

This paper describes a developed safety system for a human-centered semi-autonomous vehicle designed to detect mistakes in driver behavior where system's perception pipeline for the driving function faces an edge-case, which driver might be struggling but is not conscious of it, and then the system offers a proper alert or even issues an appropriate command. Our system applies a deep convolutional encoder-decoder network model to serve as a secondary appliance along with a vision system installed on our vehicle for the driving scene perception. In addition, we leveraged a universal in-vehicle network device to control the entire system, establish communication with each component of the system, and check all parts of the system to be properly enabled. We show that the proposed system is able to act as an effective supervisor with issuing proper steering commands and proportionate measures during driving. Our system is capable of detecting driver errors in less than 2 s using the cameras embedded in the car cabin. Thanks to the UDIAG, the system is also able to read and log all the information

mmeybodi@aut.ac.ir

about the car's ECUs. Collected data is used for a more subtle decision-making process in the system, and using this information in the future, we are able to achieve a better end-to-end model for autonomous driving.

For future work, we schedule to add an ability to monitor vehicle status on the road through drone's-eye view which has an auto-guidance system, and eke to examine and evaluate the system on today's modern vehicles with advanced navigation systems in different weather conditions.

Acknowledgments. This project was in part supported by a grant from Mehad Sanat Incorporation and Institute for Research in Fundamental Sciences (IPM).

Our team gratefully acknowledges researchers and professional engineers from Mehad Sanat Incorporation for the automotive technical consultant and offering hardware equipment.

References

1. Alessandretti, G., Broggi, A., Cerri, P.: Vehicle and guard rail detection using radar and vision data fusion. IEEE Trans. Intell. Transp. Syst. **8**(1), 95–105 (2007)
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)
3. Bertozzi, M., Broggi, A.: GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Trans. Image Process. **7**(1), 62–81 (1998)
4. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
5. Chen, P.R., Lo, S.Y., Hang, H.M., Chan, S.W., Lin, J.J.: Efficient road lane marking detection with deep learning. arXiv preprint arXiv:1809.03994 (2018)
6. Cheng, H., Zheng, N., Zhang, X., Qin, J., Van De Wetering, H.: Interactive road situation analysis for driver assistance and safety warning systems: framework and algorithms. IEEE Trans. Intell. Transp. Syst. **8**(1), 157–167 (2007)
7. Choi, J., Lee, J., Kim, D., Soprani, G., Cerri, P., Broggi, A., Yi, K.: Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment. IEEE Trans. Intell. Transp. Syst. **13**(2), 974–982 (2012)
8. Corrigan, S.: Introduction to the controller area network (CAN). Texas Instrument, Application Report (2008)
9. Fridman, L.: Human-centered autonomous vehicle systems: Principles of effective shared autonomy. arXiv preprint arXiv:1810.01835 (2018)
10. Fridman, L., Lee, J., Reimer, B., Victor, T.: 'owl'and 'lizard': patterns of head pose and eye pose in driver gaze classification. IET Comput. Vis. **10**(4), 308–313 (2016)
11. Green, P.: Integrated vehicle-based safety systems (IVBSS): Human factors and driver-vehicle interface (DVI) summary report (2008)
12. Hee Lee, G., Faundorfer, F., Pollefeys, M.: Motion estimation for self-driving cars with a generalized camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2746–2753 (2013)
13. Hoffman, E.A., Haxby, J.V.: Distinct representations of eye gaze and identity in the distributed human neural system for face perception. Nat. Neurosci. **3**(1), 80 (2000)

mmeybodi@aut.ac.ir

14. Innocenti, C., Lindén, H., Panahandeh, G., Svensson, L., Mohammadiha, N.: Imitation learning for vision-based lane keeping assistance. arXiv preprint arXiv:1709.03853 (2017)
15. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867–1874 (2014)
16. Moghaddam, A.M., Ayati, E.: Introducing a risk estimation index for drivers: a case of Iran. Saf. Sci. **62**, 90–97 (2014)
17. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **31**(4), 607–626 (2009)
18. Naranjo, J.E., Gonzalez, C., Garcia, R., De Pedro, T.: Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver. IEEE Trans. Intell. Transp. Syst. **9**(3), 438 (2008)
19. Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., Van Gool, L.: Towards end-to-end lane detection: an instance segmentation approach. arXiv preprint arXiv:1802.05591 (2018)
20. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: ENet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)
21. Smith, P., Shah, M., da Vitoria Lobo, N.: Monitoring head/eye motion for driver alertness with one camera. In: ICPR, p. 4636. IEEE (2000)
22. Soukupová, T., Cech, J.: Real-time eye blink detection using facial landmarks. In: 21st Computer Vision Winter Workshop (2016)
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
24. Varma, A.R., Arote, S.V., Bharti, C., Singh, K.: Accident prevention using eye blinking and head movement. In: Emerging Trends in Computer Science and Information Technology–2012 (ETCSIT 2012) Proceedings published in International Journal of Computer Applications®(IJCA) (2012)
25. Vicente, F., Huang, Z., Xiong, X., De la Torre, F., Zhang, W., Levi, D.: Driver gaze tracking and eyes off the road detection system. IEEE Trans. Intell. Transp. Syst. **16**(4), 2014–2027 (2015)
26. World Health Organization. Violence, Injury Prevention, World Health Organization: Global status report on road safety 2013: supporting a decade of action. World Health Organization (2013)
27. World Health Organization. Department of Violence, Injury Prevention, World Health Organization. Violence, Injury Prevention and World Health Organization: Global status report on road safety: time for action. World Health Organization (2009)
28. Wiśniewska, J., Rezaei, M., Klette, R.: Robust eye gaze estimation. In: International Conference on Computer Vision and Graphics, pp. 636–644. Springer, Heidelberg (2014)

mmeybodi@aut.ac.ir

Author Index

A
Abadeh, Mohammad Saniee, 217
Abbas Alipour, Alireza, 322
Abbasimehr, Hossein, 188
Abbaszadeh, Omid, 290
Abdi Khojasteh, Hadi, 322
Abdollahzadeh Barforoush, A., 202
Afsharchi, Mohsen, 248
Afshoon, Maryam, 1
Alagoz, Serhat Murat, 121
Amintoosi, Haleh, 311
Ansari, Ebrahim, 24, 322
Ari, Ismail, 121
Atani, Reza Ebrahimi, 59

B
Bakhshayeshi, Sina, 59
Bakır, Mustafa, 121
Bigham, Bahram Sadeghi, 13
Bohlouli, Mahdi, 1

D
Darafarin, Babak, 161
Darikvand, Tajedin, 1

E
Ebrahimpour-Komleh, Hossein, 89
Eivazpour, Z., 299
Ekramifard, Ala, 311

F
Farzaneh, Hasan, 59
Fatemi, Seyed Mohsen, 226

G
Gazerani, Vahid Gheshlaghi, 142

H
Hasheminasab, Zahir, 248
Hosseini, Seyed Mohsen, 226

I
Islam, Md.Rafiqul, 175

J
Jalili, Mahdi, 105
Jalilian, Azadeh, 24

K
Kamandi, Ali, 44, 226
Kargar, Bahareh, 142
Kavousi, Kaveh, 130
Keshavazi, Amin, 1
Keyvanpour, Mohammad Reza, 299
Khan, Rafflesia, 175
Khanteymoori, Ali Reza, 290
Khastavaneh, Hassan, 89
Khodabakhsh, Athar, 121

M
Mansoori, Fatemeh, 130
Mazaheri, Samaneh, 13
Meybodi, M. R., 202
Mirehi, Narges, 274
Moeini, Ali, 44
Mohammad Ebrahimi, A., 202
Mohammadi, Mehrnoush, 105
Momtazi, Saeedeh, 202

© Springer Nature Switzerland AG 2020
M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 337–338, 2020.
https://doi.org/10.1007/978-3-030-37309-2

mmeybodi@aut.ac.ir

Monsefi, Amin Karimi, 161
Moradi, Parham, 105

N
Narimani, Zahra, 24
Nazari, Mousa, 76

P
Pashazadeh, Saeid, 36, 76
Pazoki, Roghayeh, 238
Pishvaee, Mir Saman, 142

R
Rahgozar, Maseud, 130
Rapp, Reinhard, 258
Razzaghi, Parvin, 238, 322

S
Saber Gholami, M., 202
Sajedi, Hedieh, 217
Seno, Amin Hosseini, 311
Shabani, Mostafa, 188
Shabankhah, Mahmood, 44, 226
Sharifi, Zaniar, 248
Soltani, Reza, 36
Soltanian, Khabat, 248
Steiner, Petra, 258

T
Taghizadeh, Saeed, 44
Tahmasbi, Maryam, 274
Targhi, Alireza Tavakoli, 274
Tazehkand, Leila Namvari, 36

Y
Yavary, Arefeh, 217

Z
Zakeri, Behzad, 161

mmeybodi@aut.ac.ir