International Journal of Interactive Multimedia and Artificial Intelligence

March 2020, Vol. VI, Number 1
ISSN: 1989-1660

unir LA UNIVERSIDAD EN INTERNET

"As complexity rises, precise statements lose meaning and meaningful statements lose precision"

Lotfi A. Zadeh

Special Issue on Soft Computing

INTERNATIONAL JOURNAL OF INTERACTIVE MULTIMEDIA AND ARTIFICIAL INTELLIGENCE
ISSN: 1989-1660 –VOL. 6, NUMBER 1

Editor's Note

The International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI) covers all types of Artificial Intelligence (AI) research. The effort of reviewers, editors and authors have made the journal's quality to go up every year. Last year, the goal of being included in the Journal Citation Reports index was achieved [1]. It is totally clear to me that the journal could not have grown up that much without all the people that supports the journal. These include the abovementioned editors, reviewers and, specially, the authors, who trust and support the journal with their high-quality work. Let's continue this trail and keep growing in order to make IJIMAI the influential AI journal that it deserves to be.

Last year, a set of very interesting special issues were published in this journal. Some of them are, for instance, the Special Issue on Uses Cases of Artificial Intelligence, Digital Marketing and Neuroscience [2], the Special Issue on Artificial Intelligence Applications [3] or the Special Issue on Big Data and Open Education [4]. This time, for the first issue of the year, we are glad to present a Special Issue on Soft Computing. Soft Computing is an AI branch that focuses on solving problems that have incomplete, inexact or fuzzy information. In other words, Soft Computing area includes algorithms and methods that are typically used when the imprecision or lack of the dealt data make other type of methods to become useless. Deep Learning, Machine learning and Fuzzy Systems related methods have achieved really good results even when the available data is not as good as desired. This success has converted the Soft Computing area in one of the most important ones inside the AI field.

This special issue's goal is to reunite some of the most recent research on the Soft Computing area. The selected research covers different aspects and problems on the AI area in an effort to provide a clear overview of the state of the art on the topic. Concretely, the research included in this issue is described below.

In [5], Borhani and Ghasemloo, provide an innovative model whose purpose is to describe the urban growth. Fuzzy Logic and Artificial Neural Networks are employed by the authors in order to achieve this goal. The paper applied the designed method in a real case: the Tehran's city. Thanks to the proposed method, it is possible to know in advance how a specific city will grow. This will be critical when providing solutions that fit the citizens.

Roopa and Harish propose an interesting soft computing medical application in [6]. They use fuzzy rule networks in order to detect where a thrombus is located. For this purpose, several ECG signals taken from patients are analyzed. By studying previous detected cases, the network is capable of building a classification model that is able to automatically indicate in which artery is the thrombus located. Thanks to this method, the time required for detection is reduced and the patient can be treated in lesser time. It should be noticed that time is a critical issue in patients with thrombus. If the treatment comes late, the patient can end up dying.

Continuing the medical applications of soft computing methodologies, in [7], Devi et al. present a novel method that is capable to automatically detect skin lesions. By using an image detection technique that employs Fuzzy C-Means Clustering, they are capable of detecting the skin lesion area with a segmentation accuracy of 95.69%. This is an interesting soft computing application method that can release skin experts for finding the skin lesions by themselves. The method can do it automatically and they can focus on applying the cure. The time saving in the skin lesion finding task can be employed in assisting more patients.

In [8], Selim et al. propose an interesting application of Soft Computing for energy stability purposes. Concretely, they use fuzzy logic systems in order to detect the most efficient way of distributing energy generators on energy distributions systems. By finding the best place to put generators, it is possible to maximize the voltage stabilization. Authors compare the proposed method with others and find out that the novel developed method that they designed provides really good results.

Continuing on the electronical and electrical Soft Computing application line, Kumar et al. provide a solution to detect faults that occur on a 6-phase transmission line [9]. The developed method can successfully discriminate, classify and locate these elements. Authors have also compared their approach with other ones that already exist in the literature. They found out that their Soft Computing novel designed application provides the most robust solution to the tackled problem.

Another important Soft Computing application is shown on [10] by Wu et al. They study the use of different learning methods for classifying plantar pressure images that lack information. Thanks to this, it is possible to classify the images even when the amount of data information is low. Authors found out that a convolutional neural network provides better results in this problem than the other tested methods.

In [11], Mohammadpoor et al. propose a really interesting application of Soft Computing in the agricultural area. They employ the C-mean Algorithm in order to build a classifier that is able to detect leaves that suffer from a concrete disease: the grape fanleaf virus. Thanks to this method, it is possible to find out in short time which plants have the disease and eradicate it before the whole plantation is infected and all the production is lost. Carrying out this process using a computational system that analyses images is faster than having to carry out this process manually.

Bobadilla et al., in [12], propose the use of neural networks in order to create a recommender system from big databases. Authors propose a novel method that is able to carry out data reduction in order to design a recommender system that is capable of providing efficient solutions. They have tested the developed system with MovieLens and FilmTrust databases. Nevertheless, the proposed method can be applied to any recommender system, since it does not rely on the database topic.

Following with neural networks approaches, Maheshan proposes the use of a convolution neuronal network in order to detect the eyes' sclera [13]. Authors apply this method over the Sclera Segmentation and Recognition Benchmarking Competition (SSRBC 2015) datasets, which comprises of 734 different images. The developed method can help to create a reliable eye recognition system. Biometric applications are becoming one of the most interesting options in security applications. Among all the approaches, iris recognition is one of the most interesting options due to the fact that there are not two humans with the same iris. As authors have stated, Soft Computing can become quite useful in this area.

In [14], Saleem et al. use deep neural networks in order to enhance speeches that are recorded in noisy environments. Thanks to this, authors can effectively remove noise and improve the voice signal. Consequently, it is easier to know what is being said. A deep neural network implementation is used to obtain the required parameters of the ideal binary mask (IBM) classification function. Thanks to the correct use of the parameters, authors demonstrate that the function can effectively retrieve and improve the speech. The developed method is quite useful in cases where the parameters of

DOI: 10.9781/ijimai.2020.02.003

the algorithms are not clear. Therefore, by applying the developed approach the parameters are automatically set, and users do not have to carry out any testing task.

In [15], Hans et al. present novel Binary Multi-Verse Optimization (BMVO) approaches that can be used in feature selection. BMVO methods work resembling the physics' multi-verse theory. Concretely, they imitate the way that several universes interact among them. The main goal of the proposed approach is double. First, authors manage to provide a reliable method for removing irrelevant and redundant data. Thanks to this, the posterior learning method will be more efficient. Second, the presented approach can improve the classification accuracy that BMVO approaches normally have.

Following the bio-inspired optimization line, in [16], authors compare the ant-colony optimization algorithm (ACO) with K-Means Clustering approaches in solving one specific problem: Jobs Scheduling and Energy Optimization Model in Internet of Things (IoT). IoT is present in more and more products. Therefore, there is a need of methods, like the presented one, that is capable of efficiently manage how items work and interact.

In [17], Manju et al, present a multi-layer neural network solution for imbalanced dataset in Internet traffic classification. In order to test the accuracy of the proposed method, authors use the Cambridge dataset which consists of 248 features spread across 10 classes. Their method obtains a 99.08% of accuracy value for this highly imbalanced dataset (standard), which is the main goal of the method.

In [18], Borhani et al. present a multi-criteria optimization method for calculating flying routes in large-scale airlines. In order to achieve this goal, multi-objective genetic algorithms are applied over flights intelligent spatial information. The proposed method is tested over the Iran airline traffic patterns that were made on 2018. Thanks to the proposed approach, 50.8% of air routes were decreased. Therefore, the presented method is capable of reducing flight time and, consequently, it reduces the planes energy consumption. Thanks to this, the flights companies can gain benefits and the amount of fuel consumed is reduced.

Finally, in [19], Fong et al. present a paper whose main purpose is to aid in the recent coronavirus outbreak. Concretely, they propose a methodology for training forecasting models with small datasets. The method has three main advantages. First, it is capable of augmenting the initial existing data. Second, it includes a panel selection to select the best model. Finally, it tunes the parameters in order to reach the highest possible accuracy. The proposed methodology has been tested over data related to the 2019-nCoV disease outbreak.

Dr. Juan Antonio Morente-Molinera

References

[1] E. Verdú and R. González, "Editor's Note", vol. 5, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, n. 7, pp. 4-5.

[2] F. Mochón and A. Baldominos, "Editor's Note", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, n. 6, pp. 4-5.

[3] F. Mochón, "Editor's Note", International Journal of Interactive Multimedia and Artifical Intelligence, vol. 5, n. 4, pp. 4-5.

[4] D. Burgos, R. Nikolov, C. M. Stracke, "Editor's Note", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, n. 2, pp. 4-5.

[5] M. Borhani, and N. Ghasemloo, "Soft Computing Modelling of Urban Evolution: Tehran Metropolis", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 7-15.

[6] C. K. Roopa and B. S. Harish, "Automated ECG Analysis for Localizing Thrombus in Culprit Artery Using Rule Based Information Fuzzy Network", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 16-25.

[7] S. S. Devi, N. H. Singh, and R. H. Laskar, "Fuzzy C-Means Clustering with Histogram based Cluster Selection for Skin Lesion Segmentation using Non-Dermoscopic Images", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 26-31.

[8] A. Selim, S. Kamel, L. S. Nasrat, F. Jurado, "Voltage Stability Assessment of Radial Distribution Systems Including Optimal Allocation of Distributed Generators", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 32-40.

[9] N. A. Kumar, C. Sanjay, and M. Chakravarthy, "Mamdani Fuzzy Expert System Based Directional Relaying Approach for Six-Phase Transmission Line", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 41-50.

[10] Y. Wu, Q. Wu, N. Dey, and R. S. Sherratt, "Learning Models for Semantic Classification of Insufficient Plantar Pressure Images", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n.1, pp. 51-61.

[11] M. Mohammadpoor, M. G. Nooghabi, Z. Ahmedi, "An Intelligent Technique for Grape Fanleaf Virus Detection", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 62-67.

[12] J. Bobadilla, F. Ortega, A. Gutiérrez and S. Alonso, "Classification-based deep neural network architecture for collaborative filtering recommender systems", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 68-77.

[13] M. S. Maheshan, B. S. Harish, and N. Nagadarshan, "A Convolution Neural Network Engine for Sclera Recognition", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 78-83.

[14] N. Saleem and M. I. Khattak, "Deep Neural Networks for Speech Enhancement in Complex-Noisy Environments", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 84-90.

[15] R. Hans, and H. Kaur, "Binary Multi-Verse Optimization (BMVO) Approaches for Feature Selection", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 91-106.

[16] S. Kumar, V. Kumar Solanki, S. Kumar Choudhary, A. Selamat, R. González Crespo, "Comparatively Study on Ant Colony Optimization (ACO) and K-Means Clustering Approaches for Jobs Scheduling and Energy Optimization Model in Internet of Things (IoT)", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 107-116.

[17] N. Manju, B. S. Harish and N. Nagadarshan, "Multilayer Feedforward Neural Network for Internet Traffic Classification", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 117-122.

[18] M. Borhani, K. Akbari, A. Matkan, M. Tanasan, "A Multicriteria Optimization for Flight Route Networks in Large-Scale Airlines Using Intelligent Spatial Information", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 123-131.

[19] S.J. Fong, G. Li, N. Dey, R. Gonzalez-Crespo, and E. Herrera-Viedma, "Finding an Accurate Early Forecasting Model from Small Dataset: A Case of 2019-nCoV Novel Coronavirus Outbreak", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 6, n. 1, pp. 132-140.

TABLE OF CONTENTS

OPEN ACCESS JOURNAL

ISSN: 1989-1660

The International Journal of Interactive Multimedia and Artificial Intelligence is covered in Clarivate Analytics services and products. Specifically, this publication is indexed and abstracted in: *Science Citation Index Expanded, Journal Citation Reports/ Science Edition, Current Contents®/Engineering Computing and Technology*.

Soft Computing Modelling of Urban Evolution: Tehran Metropolis

Mostafa Borhani*, Nima Ghasemloo

Shahid Beheshti University, Quran Miracle Research Institute, 1983963113, Evin, Tehran, I.R. (Iran)

Received 14 August 2018 | Accepted 15 February 2019 | Published 1 March 2019

unir
LA UNIVERSIDAD EN INTERNET

Abstract

Exploring computational intelligence, geographic information systems and statistical information, a creative and innovative model for urban evolution is presented in this paper. The proposed model employs fuzzy logic and artificial neural network as forecasting tools for describing the urban growth. This dynamic urban evolution model considers the spatial data of population, as well as its time changes and the building usage patterns. For clustering the spatial features, fuzzy algorithms were implemented to represent different levels of urban growth and development. Then, these fuzzy clusters were modeled by the multi-layer neural networks to estimate the urban growth. Based on this novel intelligent model, the current state of development of Tehran's population and the future of this urban evolution were evaluated by empirical data and the achieved outcomes were detailed in qualitative charts. The input data-set includes four censuses with five-year intervals. Tehran's demographic evolution model forecasts the next five years with an overall accuracy of 81% and Cohen's kappa coefficient up to 74% beside the qualitative charts. These performance indicators are higher than the previous advanced models. The primary objective of this proposed model is to aid planners and decision makers to predict the development trend of urban population.

Keywords

Artificial Neural Network, Geographic Information System, Urban Development Models, Soft Computing, Fuzzy Logic, Spatial Information Systems.

DOI: 10.9781/ijimai.2019.03.001

I. Introduction

THE population growth and the migration of villagers to urban areas change the urban land use rapidly [1] and this paper models this urban evolution. According to United Nations (UN) reports, until 2030, about 60% of the world's population will be living in cities, and the average age of residents in Tehran, the metropolis chosen as a case study of this manuscript, would be 40 years old and the urban area in that year will be 1.2 billion square kilometers. Consequently, the spatial patterns of its urban areas will change dramatically by their boundaries. Because of the permanent changes in their shape and structure, modeling the evolution of urban areas is one of the most complex issues of research [2].

The decision-making process in urban environments requires a direct participation of a wide range of various aspects and a high level coordination among them. To have this process, governments typically conduct censuses. Demographic data is provided or reviewed at long intervals, such as five or ten years. In this range, the population will be assessed through non-qualitative and approximate estimations like the Average Migration Rate (AMR) or even more accurate statistical data such as Birth Rate (BR). Since the results of demographic estimates are chiefly inaccurate, and one cannot use them for providing an estimation of population dispersion, it is necessary to offer a model for appraising urban evolutions which also affect the decision-making process [3].

Typically, multi-centered contemporary cities, unlike old single-centered towns, lead to the complex evolution model. Changes in urban dispersion and land use in metropolitan areas such as Tehran demand traffic planning and high-quality life and environment, therefore, due to issues such as the increased density and lack of useful space, this requirement is becoming more apparent every day [4]. The non-managed urban development processes like for example the expansion of transportation infrastructure, industrial structures, and other construction areas affect the environment and lead it to an unsustainable development [5], and once again an urban evolution model is needed here.

Modeling and simulating complex dynamic systems like urban areas could be done by using geographical data as well as computational intelligence techniques such as fuzzy logic and useful Artificial Neural Networks (ANNs). Computational intelligence algorithms are perfectly capable of determining the pattern of changes in land and land use in non-parametric methods and places with a high degree of heterogeneity [4]. In this paper, through fuzzy boundaries [6] a method for displaying spatial categories has been offered. This fuzzy clustering provides a better profile for any spatial unit. Moreover, neural networks [7] were used for forecasting complex nonlinear issues regarding demographic evolution. Neural networks are stable against noise, and they are quite capable of automation. Also, they have no assumption about the nature of the data distribution [8]. In this study, computational intelligence techniques were used for modeling urban growth and development [7] and also by assessing the urban population dispersion through spatial data systems as well as satellite images [9], joint time-location analysis and the prediction model of urban development trend were achieved. Finally, a method for estimating urban future through neural networks, fuzzy clustering, and geographical data systems was proposed.

This proposed model has been applied and implemented specifically for Tehran city. At first, fuzzy clustering was used for the

* Corresponding author.
E-mail address: mo_borhani@sbu.ac.ir

classification of spatial units. These categories are introduced based on the statistical data of regions in Tehran, residential statistics, and their business information. By fuzzy clustering, each group is marked as an urban evolution parameter. Based on time series data, the neural network will be used for predicting the future of the city for any spatial unit.

This paper is organized as follows: in section II, diagrams of demographic growth and computational intelligence models that focus specifically on neural networks are presented. In section III, the proposed urban evolution model is described in detail and its innovations, as well as the causes of their impacts, are explicitly stated. In section IV, the case study of Tehran Metropolis using the real data achieved from censuses in four of the past five-year periods are introduced. In section V, the results of this case study are demonstrated as qualitative maps and quantitative evaluators. Then the article ends with a conclusion in section VI and references.

II. Urban Evolution and Computational Intelligence Models

Urban growth is a complicated process, and so far different models have been introduced regarding the urban analysis and for studying the changes in the urban environment through physical, economic, and social parameters [10]. Also, many parameters with different behavioral patterns in different time-location scales have been investigated in these models [7]. From a basic look and due to differences in the definition of the city, various theories respecting the analysis and simulation of urban evolution have been offered. Also, some mathematical equations have been created for the simulation of the urban phenomenon [11]; however, the relation between these theories and models are usually weak and computational capabilities have not been used as a useful tool in these models [1]. Urban growth models are an interdisciplinary process which employs different technologies such as spatial data systems, remote sensing, urban geography, and complex theories at the same time. Concerning this city and its dynamics, geographical information systems and remote sensing are considered powerful tools for managing spatial data and extracting valuable knowledge [12] - [14]. Baseline models [15] – [17], fractal models [18], and decision trees are also being used for modeling urban growth [19]. In the work done by Triantakonstantis, computational intelligence systems – as traditional models for analyzing spatial data – were employed for modeling the urban growth [19]. Regarding the land use, Omrani et al. used neural networks which were dynamic and had a discrete time – location [20]. Leo and Lodrope employed neural networks to detect new urban regions through the images of satellite sensors [21]. Their results indicated the value of neural networks in recognizing the changes. By using geographical information systems as well as artificial neural networks, Pichanoski et al. created a model for determining the land transformation based on social, political, and environmental parameters which are able of predicting the urban changes [22]. Moreover, Pichanoski et al. introduced an urban evolution model based on neural networks, geographical information system, and remote sensing. They employed a multilayered perceptron neural network, satellite images taken from 1988 and 2000, as well as social, economic, and environmental variables to estimate the expansion of metropolitan areas [23]. The combination of neural networks and fuzzy logic with experimental, conditional, and statistical methods lead to better results. For example, Guan and Clark used a dependent cellular automation model based on neural network backpropagation algorithm to simulate and forecast Beijing's evolution [24]. Dependent upon socio-economic data, this model could predict the future requests for urban space.

The previous studies suggest that in most cases, computational intelligence systems provide better natural and quantitative results than statistical methods. It is because the uncertainty principle is taken into account in this kind of methods. In most of these studies, the satellite images of remote sensing have been used and artificial neural networks, as well as a cellular automation model, were employed for simulating the urban evolution phenomenon. Regarding the neural network, time series data, satellite images, and some of the social and economic data were stored in spatial data systems and then neural networks were employed for predicting the urban evolution.

There are two main differences between the proposed method and previous studies. First of all, instead of using cellular raster data like the cellular automation model, we used statistical census analyses based on the descriptions of the urban situation, topographic information, fault, the city's green space, the distance of regions from main roads, and the distance from downtown in each spatial unit. Afterwards, we employed artificial neural networks to predict the future condition of the city. In the previous work, in the pixel based methods, the effects of other regions cannot be determined in a region for population growth in this region. The second distinction between this model and the works of the past is the usage of possibility theory and the correlation between data. In fact, in our proposed model, each region has not been investigated in separate pixels. They have been dealt with as fuzzy clustered areas with spatial information. The real changes in this paper have been studies based on time and the neighbors of each sector.

III. The Proposed Urban Evolution Model Based on Soft Computing

In this section, simultaneous analysis of the spatial and temporal data based on computational intelligence and spatial data systems for modeling urban evolution is offered. This new method has been designed based on clustering areas of study and then artificial neural networks. The structure of the proposed method for urban growth model is depicted in Fig. 1.

In this method, spatial entities will be grouped by attributes. These attributes are population, population density, land use, building usage, proximity to downtown, distance from highways, distance from commercial areas, the slope of the region, and fault map.

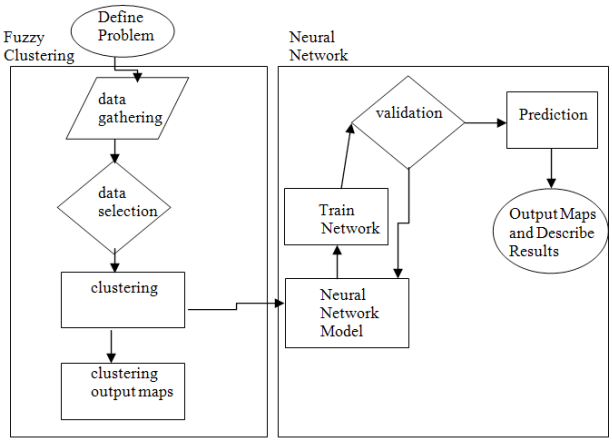


Fig. 1. The structure of the proposed method for urban growth model.

At first, the spatial reference was determined and then census and statistical data were collected. The census data at least covers four five-year periods. It is essential to select and discriminate these variables for the clustering process. This section is related to the available data. High correlated variables should be excluded from the calculations.

Clustering is a method used for recognizing the structure of data and fuzzy clustering as a proper spatial analysis is being used in

different practices. The fuzzy C-Means clustering algorithm is one the most popular clustering algorithms [6], which have also been utilized in this study. In fuzzy C-Means algorithm, data is assigned to specific categories based on similarity and proximity.

In clustering algorithms – such as K-Means – each data is allocated to only one category, however, in fuzzy clustering, each data with a degree of membership is assigned to different groups. This membership grade for each class is a number between 0 and 1. The closer the features of data to a particular category, the closer this number to 1 [25]. Wijayanto et al. used fuzzy clustering for modeling based on the mass of population [26]. Dimartinio et al. employed fuzzy clustering to introduce a fuzzy index to forecast spatial-temporal features [27]. In fuzzy clustering, every category has a center that indicates an object from that class. It is beneficial to use fuzzy clustering because these objects can be assigned to several categories with different membership degrees. On the contrary, in standard clustering, every object belongs to only one class. This indicates the level of membership value; i.e., how close the member is to the center of the categories. If the membership degree of an object relative to a class is 95%, it means that object belongs to that class; on the other hand, if the membership degree is 5%, it means that object has the least likeness to that class.

The algorithm of fuzzy clustering is like the classic average algorithm. In This method, the number of clusters is defined. The goal function of this algorithm is described as follows:

$$J = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m} d_{ik}^{2} = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m} \| x_k - v_i \|^2 \quad (1)$$

In this function, m is a real number that is greater than one. In most cases m is 2. X_k is the k_{th} sample and V_i is the center of cluster i. U_{ik} defines the membership degree of sample i relative to the class k. the sign ||*|| is the degree of samples to a center of cluster. With values u_{ik}, a matrix can be defined, that has c rows and n columns. The numbers in this matrix are between 0 and 1, but the sum of the numbers in each column is 1:

$$\sum_{i=1}^{c} u_{ik} = 1, \forall k = 1, \ldots, n \quad (2)$$

Using the above condition and minimizing the goal function, the center of cluster and the membership degree of sample i relative to the class k are computed by below functions:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)}} \quad (3)$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} x_k}{\sum_{k=1}^{n} u_{ik}^{m}} \quad (4)$$

The steps of this algorithm are described as follows:

- Define the first value for c, m
- Calculate the center of clusters
- Calculate the membership to each class
- If ||Ul+1-Ul||≤ε, then the algorithm is finished, else come back to step 2 [28].

In this paper, fuzzy clustering through C-Means algorithm was used for creating the categories which determine both the urban degree for each region and a particular grade of urban growth. These classifications show the evolution of an area in a populated urban area. In any spatial reference, the membership degree for each class for every five years will be determined. As a result, the urban condition of each spatial unit will be determined based upon that particular category.

Changes in the membership degree of spatial units that occur in different decades may change their categories. For instance, a region might belong to a category with a meager population growth whose membership degree over a 5-year period is 58% but in the next 5-year period, the membership level of this particular region would be 52%, and it would belong to a category with a small population growth. Changes in membership degree might happen in regions that their categories never alter. For example, an area belongs to a minimal demographic group whose membership degree is 88% over the first five years, and then it becomes 65% in the next five years, and lastly, it turns to 52% in the last five years. Therefore, this region is always in a category with a tiny population. Fuzzy clustering can demonstrate the decrease in membership degrees in these decades, changing from an area with a very low population density into another category. As a result, we will be able to get a better perception of the public profiles of a region, and then we can easily describe these predicted changes.

It is because these existing entities can be placed in different categories and their changes become evident over time. Once the process is done, the results will be recorded in a database, and then they will be displayed through spatial data systems, and we will be able to track the trend of each category [29]. Afterwards, this proposed structure will forecast and study the urban evolution – right after the fuzzy clustering of the place is determined. The prediction of urban growth in each spatial unit (cluster) will be made through neural networks with time series data. In the proposed structure, time series data will be improved by using spatial relations. Each unit shares borders with other units, and this identifies the same relationships between the characteristics of the groups.

It is expected that these changes in adjacent areas in a region affect another area. As a result, it is impossible to predict the growth state of a zone solely based on time series data. Therefore, in this paper, the urban evolution was identified based on the correlations between one region and its adjacent areas. Spatial data systems were used for determining the border of each region with its neighbors and also for measuring the boundary length. In this method, the new percentage value for each area is determined based on these boundaries. For instance, a region belongs to a tiny population in the neighborhood of the border, 32% is related to a small population, 45% belongs to a small population, and 23% belongs to an average population. Although the boundary length is not the only parameter that is needed to be identified for spatial interference, we just used this setting for simplification.

Time series data for each spatial unit and the adjacent units will be put into the input tables of neural networks. In this method, the neural network will be trained for learning the changes in a region and also identifying the impact of adjacent areas on that region. This structure uses these inputs, the history of spatial units, and their neighbors to predict the condition of that area in the future. A set of parameters for each neural network must be set.

Artificial neural network is a non-linear model that acts similar to a human neural system. Each ANN consists of a series of nodes and weighted connections between them [30].

A schematic diagram of the structure of an ANN is shown in Fig. 2 [30]. Each node in an ANN, receives an input from neighboring node, performs some calculation and sends the results to the other nodes as an output

In fact nodes are connected by weighted connectors to different

layers in the ANN. In ANN there are different layers that can be generally divided into three groups of input layers, hiding layers and output layers [30].

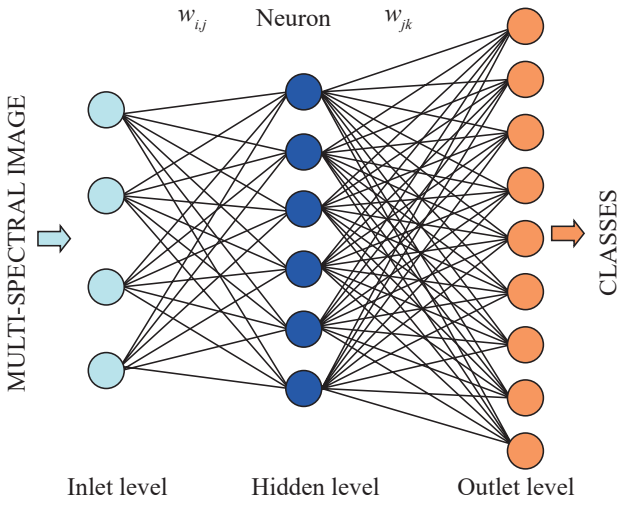


Fig. 2. Schematic diagram of the structure of an ANN [30].

ANN consists of two stages in classification, training and recalling [31]. In the training stage, an operation is being conducted to improve the connection weighting coefficients through some iteration processes. In this training stage the known inputs and outputs are used, and consequently is called Back-Propagation Supervised Classification [33]. This method is a gradient iteration algorithm that is being used to minimize the error as is shown in equation 5.

$$E = \frac{1}{2}\sum_{j=1}^{L}(d_j - o_j^M)^2 \tag{5}$$

where, d_j and o_j^M are the desired output and ANN output respectively. Based on the results of equation, the amount of correction for each weight can be calculated [32].

$$\begin{cases} \Delta w_{i,j} = -\eta \dfrac{\partial E}{\partial w_{i,j}} \\ \Delta w_{i,j}(t+1) = \Delta w_{i,j} + \alpha \Delta w_{i,j}(t) \end{cases} \tag{6}$$

Here W is the weight between two nodes and η is a positive constant parameter that is being used for adjustment and is called training rate. α is a momentum factor (usually called stability factor) and can take a value between 0 and 1 and t is the number of iterations [32]. In the recalling stage, ANN works according to the trained weighting coefficients, and conducts interpolations and extrapolations. These parameters include the number of intermediate layers, the number of neurons in each layer, training rates, momentum, the number of repetitions, and activity functions. It is important to adjust each parameter correctly because the ultimate accuracy and the acceptance of any neural network depends on this action. One of the privileges of ANN method compared to traditional statistical methods is that the networks are free in distribution i.e. the training and recalling are depending on the linear combination between data patterns and are independent of input data [30]. However the reasons for the success of ANN in classification can be summarized as: 1) there is no need for pre-assumption in data distribution 2) it is faster compared to the other methods 3) it permits user to make use of the initial knowledge regarding classes and possible limitations 4) the method is able to manage the spatial data from multi-sources and can achieve their classification results equally [30].

Concerning neural networks, user experiences and right and wrong actions are considered necessary elements. Afterwards, some statistical indicators such as least squares error and correlation coefficient will be evaluated, and their acceptance rate will be identified. Predictive maps will be designed in the next step. In other words, the outputs of the neural network will be close to the reference outputs.

IV. The Case Study of the Proposed Model Regarding Tehran Metropolis

The proposed model has been implemented for modeling the urban evolution of Tehran Metropolis in all 22 zones of the municipality. Tehran's regions include small areas which are themselves like a city. The capital of Iran, Tehran, has been chosen as a case study because it possesses some particular features such as the high focus of industry in the city, manufacturing activities, services, and workforce. The area of Tehran is about 730 km^2, and its population is about 8 million, which is divided into 22 local zones. Fig. 3 demonstrates Tehran's geographical location.

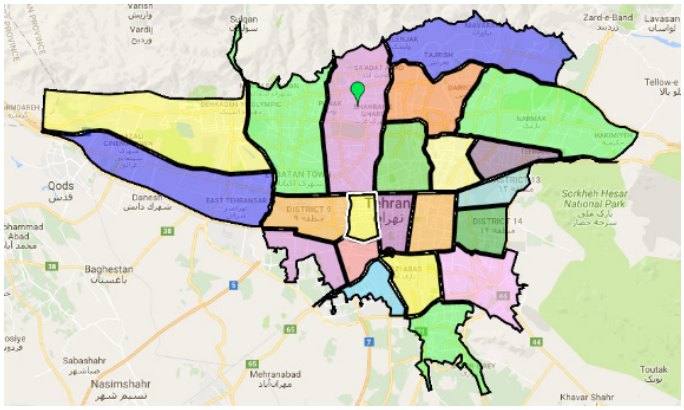


Fig. 3. Case study, Tehran.

A. Data

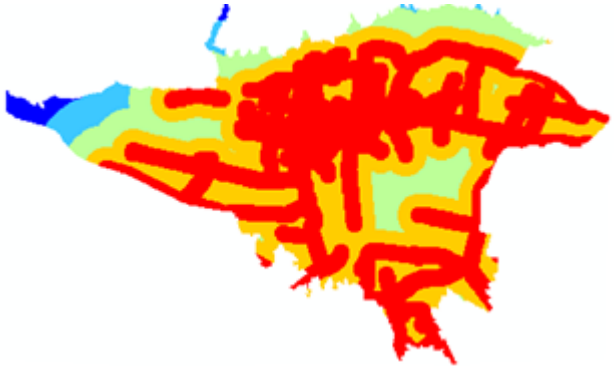
Due to the existence of control information, the spatial unit reference in this research was considered the municipalities. The data of Iran's statistical services centers include building usage and the population from 1996 to 2011. Statistical information relative to 2001 was collected through interpolation. Census variables are being used for defining the "Urban Evolution Model," while it is possible that other variables will be studied in future articles.

The data include four consecutive five-year censuses, and each census comprises eight parameters as follows: 1. Population, 2. The number of residential usages, 3. Distance from downtown, 4. Distance from fault, 5. Distance from highways, 6. Green space, 7. Industrial practices, and 8. The slope map. Some of them are shown in Fig. 4. Through these variables and by population, population growth rate, and building usages, the profile of municipalities that are well suited to the public profile will be acquired. For instance, the high number of buildings and a significant population indicate the downtown.

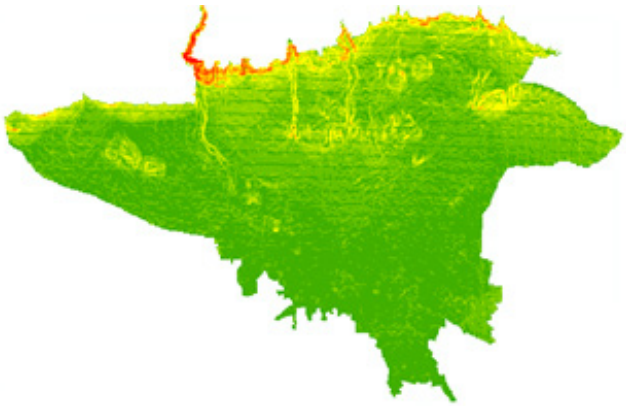
B. The Results of the Implementation of Clustering Analysis

The fuzzy C-Means algorithm was used for the required clusters. Clustering was done for every five years, and the categories suggest the urban population grade for each region. Five categories named very low population, low population, average population, high population, and excessive population were considered.

(a)

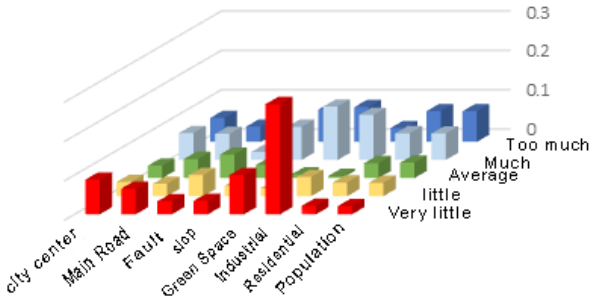


(b)

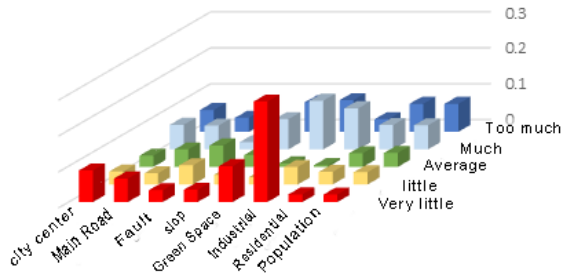


(c)

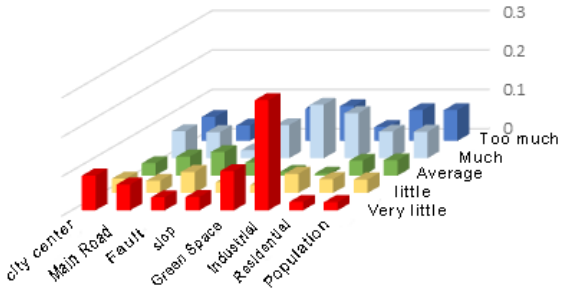
Fig. 4. (a) Distance from fault, (b) distance from main roads, (c) Tehran's slope maps A-C.



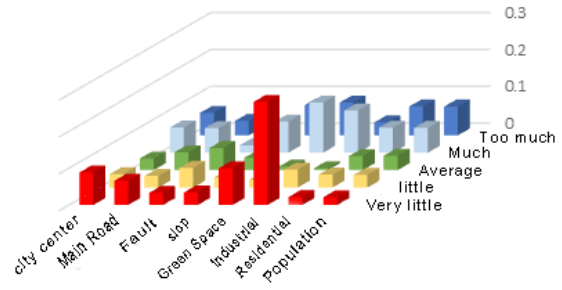
(a)



(b)

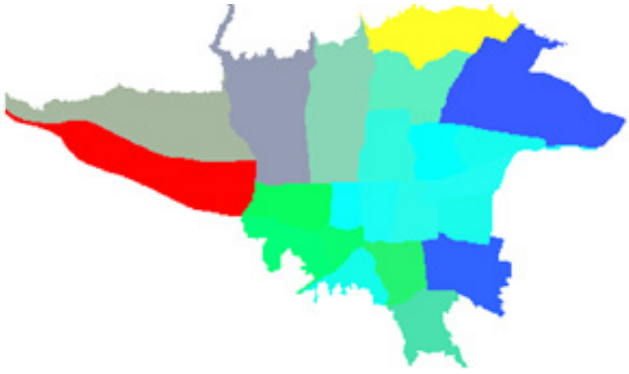


(c)



(d)

Fig. 5. The category centers in four five-year periods: (a) 1996, (b) 2001, (c) 2006, and (d) 2011.



(a)

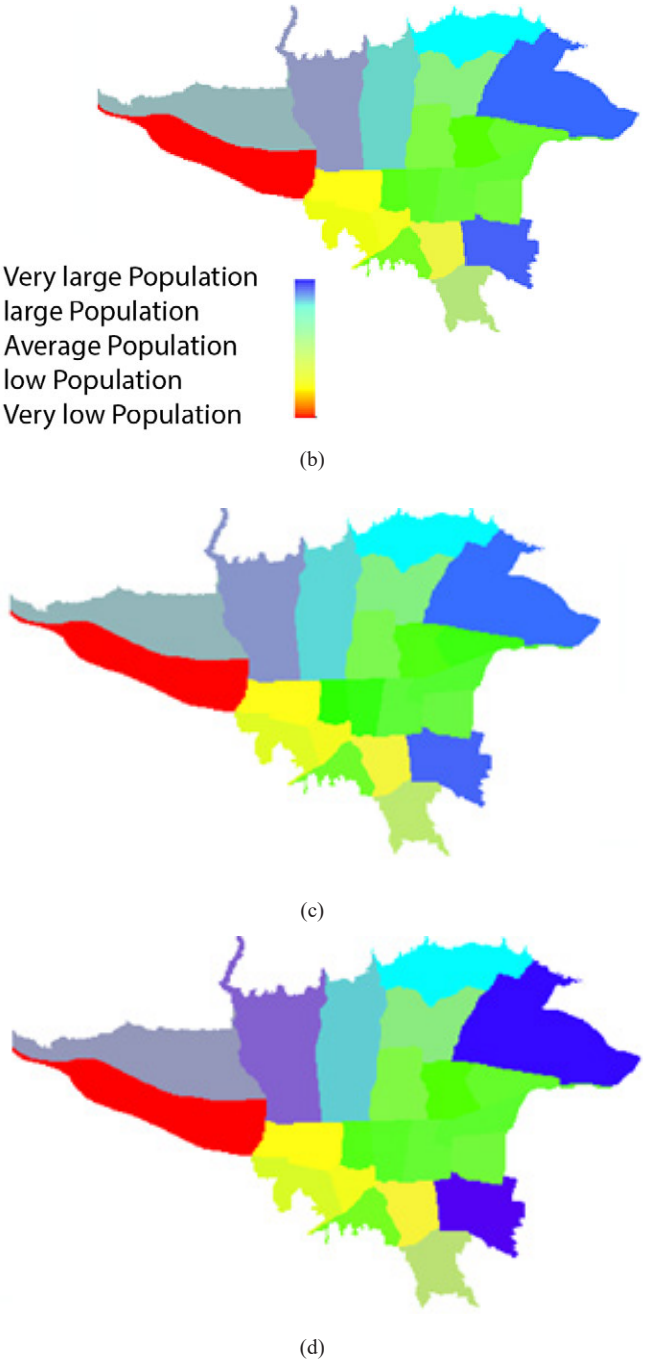


Fig. 6. Clustering Tehran's regions using C-Means algorithm in (a) 1996, (b) 2001, (c) 2006, and (d) 2011.

Fig. 5 indicates the centers of the categories over the four 5-year periods – 1996, 2001, 2006, & 2011 – and each class were described based on the values of the group centers. As you can see, those categories with a very small population have the most industrial activities, and the highest amount of green space is perceived in the categories with a large population. Each region belongs to classes with different grades over five years; e.g., if an area belongs to the category with a very low population in 2011, its statistical values are similar to the category with a very low population. Hence, the centers of categories refer to the statistical values in every five years. Each group has been described based on its center as follows:

- Very low population: in these areas, compared to the other sectors, a low amount of people resides. These areas are far from the downtown and the main roads. There are many factories in these regions which influence its weather conditions.
- Low population: these areas have a low population and few green spaces. They are close to the downtown. Most of the offices are located there. They are almost close to the main roads.
- Average population: this category has an average population. There are few industrial regions there. These areas are near the downtown and main roads.
- High population: these areas have a quite excessive population. They are far from the downtown, and they have many green spaces.
- Very high population: these areas have many green spaces and are densely populated. Many buildings can be seen in these regions. They have few industrial zones, and they are close to the main roads.

In every five years, each region from membership degree of fuzzy C-Means algorithm will belong to one of these classes. Fig.6 shows the clustering of these membership degrees over the four five-year periods – 1996, 2001, 2006, & 2011.

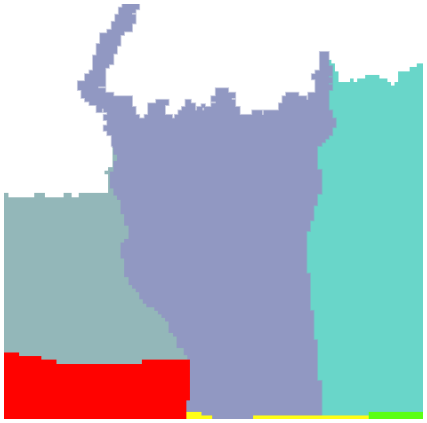
Through this clustering, we will be able to understand each spatial unit with its membership degree changes better while other clustering methods are unable to track changes. In these categories, Fuzzy Logic of the urban population for each class is determined. Through this process, the trends of each region will be identified, and then the city's future urban evolution will be simulated. For example, in Fig. 6, the increase in urban population in Tehran's northern and western regions is presented. Due to Tehran's air pollution in central areas, its residents would rather live in northern regions which have a more suitable climate. This is entirely evident in Tehran's northwest regions. For instance, 35% of the residents in Tehran's zone 5 preferred populated areas in 1996 while in 2011, with a 46-percent membership degree, it was named a populated zone. These changes are visible in Fig. 7; the color of this district changes from red (low population) to blue (high population). The northeastern regions of Tehran have a better climate; hence, many people have been living there during these years. On the other hand, the population of Tehran's central areas was decreasing, and people preferred to live in a field with average population.

C. The Implementation of Neural Networks for Prediction

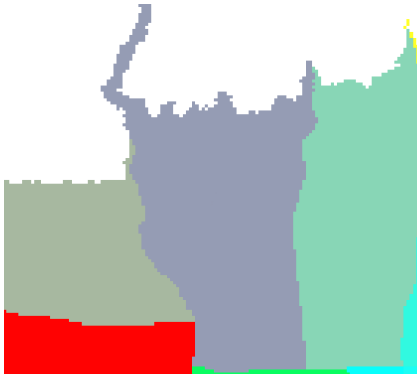
Regarding the prediction of the city's growth, the multilayer perceptron neural network was chosen and implemented. The implementation of input data has ten features of which the first 5 data include the membership degree of each region for every five years. Data sixth to tenth contain the average degree of membership of the surrounding areas for the same five years. The outputs of this model comprise 5 data that indicate the membership degree in the next five years. Consequently, the future state of the city development will be determined. 70% and 30% of the data were selected for training and testing respectively. To reach a proper network, more than 50 combinations of neural network settings were employed for trial and error. This suitable system has an intermediate layer relative to five neurons in a linear function. The amount of momentum training for two layers was considered 0.5. Based on the results, the weights were selected with the highest accuracy. For example, minimum mean squared error for training data after 1000 repetitions was 0.0001. Table I presents the architecture (i.e., the intermediate layers, conversion functions, step size, and momentum size) of the neural network which has been achieved after 1000 repetitions in the trial and error phase. Through test data, as follows, the neural network was tested. The test data were not used during training. The real data was employed to test the neural network. The reference output and system output for a part of the test data are presented in Fig. 8. The neural network was trained for determining the data, and then it got prepared for prediction.

TABLE I. The Neural Network Settings

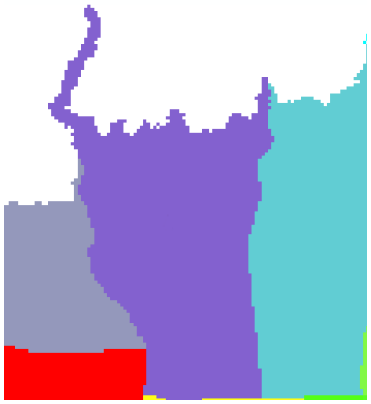
The number of hidden layers	The number of neurons in layer 1	The number of neurons in layer 2	The number of repetition	Minimum square error	Overall accuracy of test data (percent)
1	5	0	1000	0.0002	**79**
1	10	0	737	0.0001	**80**
2	10	10	507	0.0001	**80**
2	20	20	239	0.0001	**46**



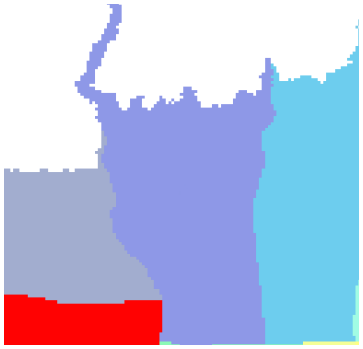
(a)



(b)



(c)



(d)

Fig. 7. The membership degree changes in Tehran's district 5 in (a) 1996, (b) 2001, (c) 2006, and (d) 2011.

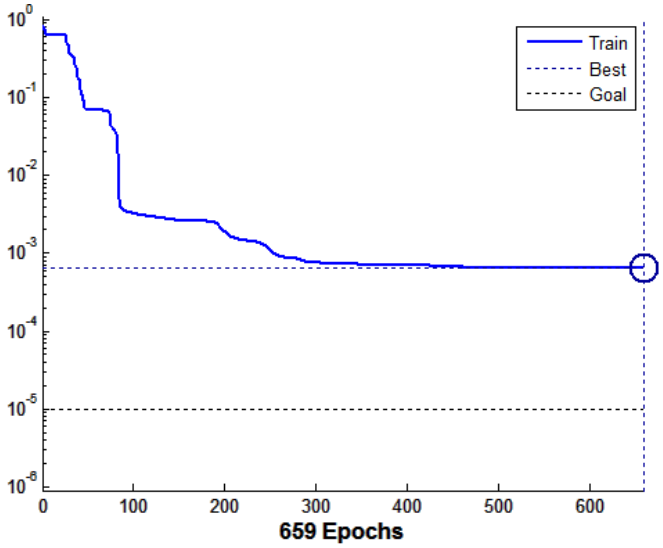


Fig. 8. The minimum square error of artificial neural network training.

D. Evaluation of Results and Discussion of Prediction

The confusion matrix was used for evaluating the quantity results gained from the prediction of the neural network. This matrix arises from data to data and known data to corresponding data comparisons of the classification results. In this matrix, the known data are represented in columns, and the data of classification results are shown in rows. Two critical parameters named overall accuracy and Kappa coefficient were used for evaluation. The overall accuracy is the average of classification accuracy that considers the proportion of data with a correct classification of the total data. In fact, by removing the effect of chance on classification, the amount of compliance with the actual data will be specified. These results can be seen in Table II.

TABLE II. using the Confusion matrix to determine the accuracy of prediction

Category	Very low	Low	Average	High	Very high	Total
Very low	2	1	0	0	0	3
Low	0	8	0	1	0	9
Average	0	0	13	0	0	13
High	0	0	0	2	2	4
Very high	0	0	0	3	5	8
Total	2	9	13	6	7	37

The data collected from 2011 was used for predicting 2016. The outcomes of the membership degrees were related to the regions, and the results are presented in Fig. 9.

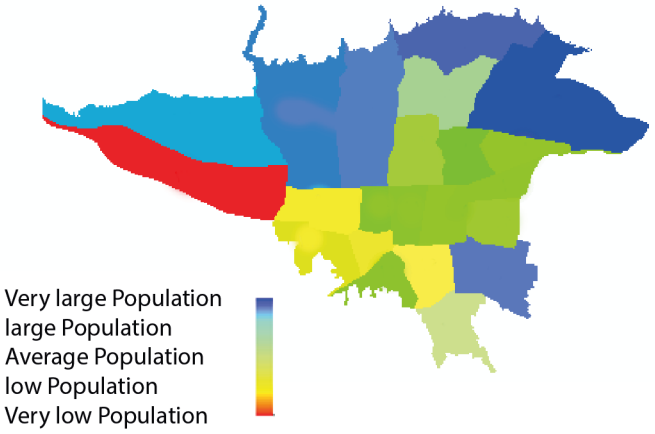


Fig. 9. Neural network prediction of population growth procedure in 2016.

E. Discussion

In this proposed method for modeling the urban evolution, fuzzy clustering and neural networks were used for analyzing and predicting the urban growth. Fuzzy clustering offers some descriptions of the urban condition of any spatial unit, and then a neural network will forecast the future state of the city. Building and population data were employed for estimating the urban growth; however, satellite images were never used.

A densely populated area could be considered a place for recruiting residents and constructing new buildings. As a result, the urban growth distribution will be specified. Through the predictive maps (Fig. 9), as you can see, Tehran's northern and northwestern area is populated, and as it is predicted, its central areas will become average or low populated areas. According to the predictions, zone 5 will have a high population, and it will require a proper planning for resource allocation.

Population growth in northern and northwestern areas of Tehran causes more traffic and effects on air pollution. In addition, population density raises the needs for resources.

According to the results of the proposed method, the reason of low population in south and southwestern areas should be taken under consideration and try to fix the defects in order to have most equally distributed population all over the city.

The proposed method has some limitations such as resolution power in the units of these areas. In the future, these studies can be useful for areas inside each region. Moreover, by using remote sensing data, we will be able to improve the capabilities of the algorithm.

V. Conclusion

Urban evolution can be modeled through fuzzy clustering and neural networks. Although the proposed method is simple, it can be adapted to instantaneous dynamic structures. Fuzzy clustering is an appropriate means and was used in this model. Consequently, each spatial unit has its profile over every five years. Fuzzy Logic was used to create a full picture of the areas studied for five years and also it presented the membership degree changes and upcoming trends for each spatial unit.

Through the neural network, the future urban state of each region was specified and, as a result, a better profile for decision-making was offered. In the future, this method with more temporal data and a completed subject database could become an efficient spatial model. Practical methods are precious regarding Tehran City because they determine the tendencies of areas and development of buildings in the future.

References

[1] E. Vaz, P. Nijkamp, "Gravitational forces in the spatial impacts of urban sprawl: An investigation of the region of Veneto, Italy", Habitat International, 2015.

[2] L. Jiao "Urban land density function: A new method to characterize urban expansion", Landscape and Urban Planning: 26-39, 2015.

[3] R. Aghataher, N. N.Samani, "Spatial Demographic Distribution Zoning of a City with Area Interpolation Method using Image-based Geo-spatial Information System-Case Study: Tehran City", Journal of Geospatial Information Technology, Iran (4): 65-82, 2016.

[4] J. Wang, G. Mountrakis "Developing a multi-network urbanization (MuNU) model: A case study of urban growth in Denver", International Journal of Geographical Information Science, 25(2): 229–253, 2011.

[5] J. Barredo, C. Lavalle, M. Kassanko,"Urban Scenario Modeling and Forecast for Sustainable Urban and Regional Planning, in GIS for sustainable development. M. Campagna, Eds., CrcPres", Taylor & Francis: 329-349, 2005.

[6] G. Grekousis, T. Hatzicheros, "Comparison of two fuzzy algorithms in geodemographic segmentation analysis" The fuzzy C-Means and Gustafson Kessel methods, Applied Geography (34): 125–136, 2012.

[7] J. Cheng, "Modeling spatial and temporal urban growth", PhD Dissertation, Utrecht University, 2003.

[8] J. D. Olden, D. A. Jackson, "Fish–habitat relationships in lakes: Gaining predictive and explanatory insight by using artificial neural networks", Transactions of the American Fisheries Society (130): 878–897, 2001.

[9] S. Mohammady, M. R. Delavar, "Urban sprawl assessment and modeling using LANDSAT images and GIS" Modeling Earth Systems and Environment 2.3: 155, 2016.

[10] O. Hamdy, S. Zhao, T. Osman, M. A.Saleheen, Y.Eid, "Applying a Hybrid Model of Markov Chain and Logistic Regression to Identify Future Urban Sprawl in Abouelreesh", Aswan: A Case Study, Geosciences, 6(4): 43, 2016.

[11] G. Grekousis, M. Panos, N. Yorgos, "Modeling urban evolution using neural networks, fuzzy logic and GIS, The case of the Athens metropolitan area", CITIES (30):193-203, 2013.

[12] S. Movahedi, M. Taleai, M. Karimi, "Development of a GIS-based model for Locating Urban Neighbourhood and District Centres based on Mixed Land Use Concepts", Journal of Geospatial Information Technology,Vol.3, No.3, autumn, 2015.

[13] J. Kubanek, E. Nolte, M. H. Taubenbock, F. Wenzel, M.Kappas," Capacities of remote sensing for population estimation in urban areas", Journal of Earthquake Hazard Impact and Urban Planning Environmental Hazards: 45-66, 2014.

[14] J. Kumar, P. K. Garg, D. Khare, "Monitoring and modeling of urbansprawl using remote sensing and GIS techniques", International Journal of Applied Earth Observation and Geoinformation, 10(1): 26–43, 2008.

[15] I. Benenson,"Multi-agent simulations of residential dynamics in the city, Computers", Environment and Urban Systems, 22(1): 25–42, 1998.

[16] J. Kerridgw, J. Hine, M. Wigan, "Agent-based modeling of pedestrian movements: The questions that need to be asked and answered", Environment and Planning B: Planning and Design (28): 327–34, 2001.

[17] F. Hosseiali, M. Azizkhani, "Developing an agent-based model for spatial simulation of pedestrian's behavior passing across the street and using the pedestrian bridges", Journal of Geospatial Information Technology, Volume 4, Number 2 (9-2016), 2016.

[18] G. Shen, "Fractal dimension and fractal growth of urbanized areas", International Journal of Geographical Information Science, 16(5):419–437, 2002.

[19] D. Triantakonstantis, G. Mountrakis, J. Wang, "A spatially heterogeneous expert based (SHEB) urban growth model using model regionalization", Journal of Geographic Information System(5): 195–210, 2011.

[20] H. Omrani, A. Tayyebi, B. Pijanowski, "Integrating the multi-label land-use concept and cellular automata with the artificial neural network-based Land Transformation Model: an integrated ML-CA-LTM modeling framework", 2016.

[21] X. Liu, J. R. Lathrop, "Urban change detection based on artificial neural network", International Journal of Remote Sensing, 23(12):2513–2518, 2002.

[22] B. C. Pijanowski, D. G. Brown, B. A. Shellito, G. A. Manik, "Using neural networks and GIS to forecast land use changes: A land transformation model", Computers, Environment and Urban Systems(6): 553–575, 2002.

[23] B. C. Pijanowski, A. Tayyebi, M. R.Delavar, M. J. Yazdanpanah, " Urban expansion simulation using geo-spatial information system and artificial neural networks", International Journal of Environmental Research,3(4):493–502, 2009.

[24] Q. L. W. Guan, K. C. Clarek, "An artificial-neural-network-based, constrained CA model for Simulating urban growth", Cartography and Geographic Information Science, 32(4): 369–380, 2005.

[25] G. S. Chandel, P. Kailash, S. Man, "A Result Evolution Approach for Web usage mining using Fuzzy C-Means Clustering Algorithm", International Journal of Computer Science and Network Security (IJCSNS) (16.1): 135, 2016.

[26] A. W. Wijayanto, A. L. Purwarianti, H. Son, "Fuzzy geographically weighted clustering using artificial bee colony: An efficient geo-demographic analysis algorithm and applications to the analysis of crime behavior in population", Applied Intelligence: 377–398, 2016.

[27] F. Di Martino, R.Mele, S.Sesa, U.Barillari, M.Barillari, "WebGIS based on spatio-temporal hot spots: an application to oto-laryngo-pharyngeal diseases", Soft Computing, vol. 20, (6): 2135–2147, 2016.

[28] R. Hathaway, J.Bezdek,"Nerf C-Means: Non-Euclidean Relational Fuzzy clustering", Pattern Recognition, 27 , 429-437, 1994.

[29] M. Borhani and H. Ghassemian, "Kernel Multivariate Spectral–Spatial Analysis of Hyperspectral Data," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2418-2426, June 2015. doi: 10.1109/JSTARS.2015.2399936.

[30] F. Caravajal, E.Crisanto, F.Aqguera, M. A. Aguilar, "Greenhouse detection using neural network with a very high resolution satellite image.", ISPRS Technical Commission II Symposium, 37-42, 2006.

[31] M. Mokhtarzadeh, M. J.ValadanZoej, "Road detection from high resolution satellite images using artificial neural networks", Int. J. of Appl. Earth Observation and Geoinformation,32–40, 2007.

[32] G. Y. Yang, "Geological mapping from multi-source data using neural networks". M.Sc. Thesis. University of Calgary, Canada, 1995.

[33] M. Borhani, Corpus Analysis Using Relaxed Conjugate Gradient Neural Network Training Algorithm, Neural Processing Letters, 2018. https://doi.org/10.1007/s11063-018-9948-8

Mostafa Borhani

Mostafa Borhani received the BSEE degree and MSEE degree from Sharif University of Technology, in 2002 and 2004, respectively and he successfully defended his doctoral thesis entitled "Hyperspectral Remote Sensing Image Classification Based On Spectral-Spatial Kernel Methods" at the Department of Electrical and Computer Engineering, Tarbiat Modares University in 2015. His research interests include machine learning, artificial intelligence, kernel methods, adaptive algorithms and joint time-frequency analysis with special focus on Holy Quranic data analysis to achieve more comprehension and understanding on this holy book. His research is conducted as faculty member of the Quran Miracle Research Institute of the Shahid Beheshti University.

Nima Ghassemloo

Nima Ghassemlo received the BS degree from Tabriz University, in 2007 and MSEE degree in 2010, respectively and he is Ph.D. candidate at the GIS and remote Sensing Center of Shahid Beheshti University since 2016.

Automated ECG Analysis for Localizing Thrombus in Culprit Artery Using Rule Based Information Fuzzy Network

C K Roopa*, B S Harish

JSS Research Foundation, JSS TI Campus, Mysuru, Karnataka State (India)
JSS Science and Technology University, JSS TI Campus, Mysuru, Karnataka State (India)

Received 28 September 2018 | Accepted 15 February 2019 | Published 25 February 2019

Abstract

Cardio-vascular diseases are one of the foremost causes of mortality in today's world. The prognosis for cardio-vascular diseases is usually done by ECG signal, which is a simple 12-lead Electrocardiogram (ECG) that gives complete information about the function of the heart including the amplitude and time interval of P-QRS-T-U segment. This article recommends a novel approach to identify the location of thrombus in culprit artery using the Information Fuzzy Network (IFN). Information Fuzzy Network, being a supervised machine learning technique, takes known evidences based on rules to create a predicted classification model with thrombus location obtained from the vast input ECG data. These rules are well-defined procedures for selecting hypothesis that best fits a set of observations. Results illustrate that the recommended approach yields an accurateness of 92.30%. This novel approach is shown to be a viable ECG analysis approach for identifying the culprit artery and thus localizing the thrombus.

Keywords

Cardio-Vascular Diseases, ECG, Identification, Classification, Information Fuzzy Network.

DOI: 10.9781/ijimai.2019.02.001

I. Introduction

The regular 12-lead Electrocardiogram (ECG) has been employed clinically for more than 100 years for finding heart diseases [1]. An ECG supervising system evaluates the ECG signals from leads incorporated in different anatomical locations of the body for evidences of ST elevation [2]. ST elevation and depression measurements from the leads are plotted in a graphical display arranged in relation to the anatomical points that are the basis of the lead signals [3]. Supervising ECG for ST segment variation is the only non-invasive procedure that facilitates the recognition and documentation of all episodes of Ischemia as replicated by ST segment changes [4].

Machine learning techniques are being increasingly adopted for automatic identification of blocks by analyzing ECG signals. Neuro-Fuzzy methods can be used to estimate the unknown noise, present in the Electrocardiogram signals [5]. Adaptive neural combined with the Fuzzy System, is used to construct a fuzzy predictor for ECG analysis. For this system, setting up parameters such as the number of Membership Functions (MF), training time period and learning algorithm are determined by the learning data [6]. Due to the altered form of ECG signals with several frequency noises and existence of several arrhythmic events in a cardiac rhythm, automated elucidation of abnormal ECG classification becomes a demanding task [7].

However, adequate Identification of Thrombus in the Culprit Artery (ITCA) in the patient before performing an angiogram is necessary for successful surgical outcome. Before designing an automatic thrombus identification system, the characteristics and properties of ECG signal have to be studied with respect to the thrombus location. Due to a wide range of patients with Myocardial Infarction (MI), the characteristic of ECG signal may vary among patients with the same heart disorder. Hence the main objectives of this paper includes: 1. Differentiating Non-Ischemic cause of ST elevation from Ischemic cause 2. Localizing culprit vessels and its segments by ECG using ST elevation myocardial infarction. Judgment of accurate location of a lesion in the coronary artery by analyzing the ECG signal is important in a typical intensive care or operating room for decision-making and further treatment of the patient. The key idea of the proposed method is to identify the culprit artery and thrombus location by eliminating other chances or causes such as Tachycardia using ST elevation in myocardial disorder. For this purpose, this article proposes a novel classification method for detecting the block in coronary artery, based on traditional conditions and criteria [9]. These conditions depend on ST Elevation or depression voltage levels in different lead systems and time intervals between points. The main advantage of the proposed method is that the cardiologist can make a quick decision to take further clinical procedures and can act as an aid for the physician (who is not specialized in cardiology) in primary health care centers.

The basic outline of the proposed methodology for detection of thrombus is as follows: ECG signal from the database is fed to the Savitzky-Golay filter [8] for smoothing. The feature parameters required to classify are measured using Stockwell Transform (starting point, peak point, end point), Nearest- Neighbor Interpolation (time interval between points) and measurement of peak amplitude (peak amplitude). The design of the overall recommended methodology is

* Corresponding author.
E-mail address: ckr@sjce.ac.in

presented in Fig. 1.

Stockwell Transform [37] is an improvisation and combination of Short-Time Fourier Transform (STFT) and wavelet transform. The main function of this method is to analyze time-frequency and this paper implements Stockwell transform for point identification in ECG signal to locate the starting point, peak point, and end point of the ECG signal. Further, Nearest-Neighbor Interpolation is used for interpolating functions of more than one variable that constructs original data within the series of distinct set of identified data points. Further, it is used for calculating the time interval between two identified points in the signal. Nearest-Neighbor Interpolation is followed by Peak Amplitude Measurement, which is used to calculate the mV range of the base to peak voltage level. Finally, the features identified in ECG signal are classified based on the traditional criteria [9] using the Information Fuzzy Network.

In summary the proposed method consists of five steps: (i) Discretization and point detection using Stockwell Transform, (ii) Feature identification based on Nearest-Neighbor Interpolation and peak amplitude measurement, (iii) Evaluation of association rules based on traditional criteria, (iv) Classification using Information Fuzzy Network and (v) Block detection.

The organization of the article is structured as follows: Section II presents some of the current research work done in the automated localization of thrombus by classification of ECG Signal. The proposed methodology is presented in section III. The empirical results and the performance assessment of the proposed method with other traditional methods are presented in section IV, followed by conclusions in section V.

II. Literature Review

Identification of Myocardial Infarction (MI) and Ischemic beat is done on pre-processed ECG signal. Pre-processing includes removing of Baseline Wander (BW) and Noise. The basic method used for BW removal is the Butterworth filter in [10, 11] and the authors in [12] used cut-off frequency depending on heart beat for 3rd order Butterworth filter. In [13] 0.3Hz cut-off frequency and in [14] authors used a cut-off frequency of 0.5 Hz. In [15], BW was eliminated using the low frequencies derivative-based filter. On the other hand [16, 17] employed moving average filtering to eliminate the BW. Further studies proposed in [30] employed median filter to estimate BW. In [25], median filter was used to remove BW by taking the difference between the residue component of Empirical Mode Decomposition (EMD) and Original signal. Discrete Wavelet Transform (DWT) has also been employed for BW elimination. Many techniques are employed in literature to lessen the noise in ECG signals. Low-pass filters are the simplest once. Authors in [15] applied low-pass filters to remove high frequency noise. Author in [8] used SG filter with order 5. While in [26] author used Butterworth filter without specifying frequency and order of the filter. Low-pass filtering is also been employed using moving average filters in [19, 26]. After pre-processing the ECG signals, feature extraction is done to find the MI and Ischemia.

The feature extraction method uses the amplitude, points, time interval, segments and frequencies of the ECG. The Fourier transform is the most important method in signal processing which signifies the quantity of the signal made up by each frequency. Similarly Wavelet Transform, Fourier Transform, Principal Component Analysis (PCA) and reconstructed phase space (RPS) theory can be used in feature extraction [47]. Features extraction from wavelet transform, wavelet db4 [31], db6 [30] and db9 [19] is also useful in the identification of MI. EMD is used for extraction of feature in [31]. Since EMD can decompose signal into different frequencies, only 2nd level component are used. However, the first level corresponds to noise.

In literature many methods have been employed for dimensionality reduction for MI detection. Forward Feature Selection (FFS) used in [32, 35] is computationally more efficient over simple filer based methods. 840 features composed of wavelet coefficients was reduced using Fuzzy rough feature selection in [44] using different level of wavelets. Particle Swarm Optimization (PSO) [32] and Bat algorithm [27] have been used in features selection. Machine learning algorithms like Random Forest and Lasso are also used in features selection. In [36], Multilayer Perceptron (MLP) eliminates irrelevant features for classification.

Selected features are used in majority of classification techniques as normal or MI/ischemia ECG signal. Thresholding method was widely used for ECG classification in [20, 22, 45, 28, 29]. Using K value as 5, KNN classifier was used in [18] to classify the signal using description found in [46]. Support Vector Machines (SVMs) have been widely used for classification of ischemia and MI with linear [17, 18, 24, 33], Radial Basis Function (RBF) [18, 21, 25, 35], polynomial [11] and exponential chi-squared kernel functions [11]. Review in [47] used artificial neural networks, fuzzy logic, rough set theory, decision trees, genetic and hybrid algorithms for classification of various heart

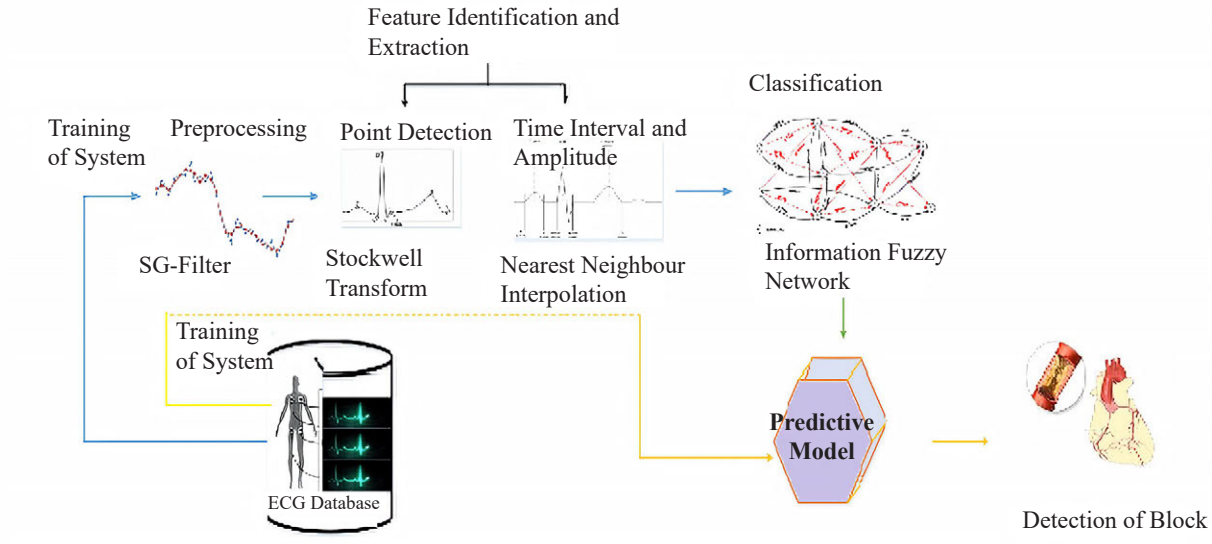


Fig. 1. Location of Thrombus in the Culprit Artery using Information Fuzzy Network.

diseases. Classifications based on Predictive Association Rules (CPAR) algorithm applied 6 association rules presented in [23] to classify ECG features. Expert knowledge is additionally been modelled with fuzzy logic in [5].

The vital role of a cardiologist is to take care of cardiac patients with utmost concerns. Thus computer aided feature extraction and analysis of ECG signals for disease identification has become a necessity. The initial step in Computer Aided Diagnosis (CAD) is the identification and extraction of the features of the ECG signals.

III. Research Methodology

The first stage in developing an automated ECG classifier and thrombus location is to extract the characteristic features such as the points (P, Q, R, S, and T), time interval and amplitude from the waveform. Time interval and amplitude is considered as the first order and hence can be derived using Nearest-Neighbor Interpolation and Amplitude measurement technique. On the other hand, point detection can be obtained using Stockwell Transform. In a typical supervised classification scheme, the features are labeled with the decision outcome. Further, the IFN classifier is applied to the data through training and testing process.

A. Identification and Extraction of Time Interval and Amplitude

In the proposed methodology, classification of the ECG signal and identification of the culprit artery with the location of thrombus is fundamentally based on some of the features obtained from the ECG signal. The features that are identified and extracted are the point's identification (starting point, peak point, end point) of P, Q, R, S, T, time interval between points and peak amplitude. For point identification, an ECG signal has to be converted into its time–frequency distribution and frequency-dependent resolution, while preserving a direct relationship with time spectrum component. For this purpose, the Stockwell Transform [37], is used. The advantage of using Stockwell Transform is that it gives an accurate Time-Frequency representation and gives information about all points present in the signal. After identifying the ECG points, the necessary time interval between the points are calculated using Nearest-Neighbor Linear Interpolation method. It performs linear polynomials to construct original data points within the series of distinct set of identified data points. Finally, the amplitude of the ECG points such as R peak amplitude and ST segment amplitude is calculated by implementing Amplitude Measurement technique [38], which uses uneven threshold process to identify the R-peak and ST segment amplitude.

B. Stockwell Transform

The Stockwell Transform provides a frequency dependant resolution spectrum. The advantage of this transform is that it maintains a straight association with the Fourier spectrum. In this work, the advantages of Stockwell Transform to identify the ECG points are exploited. Stockwell Transform is equivalent to applying several Short-Time Fourier Transform (STFT) with different sized windows. It can be examined as a frequency dependent STFT or a phase corrected wavelet transform. In addition, it permits accessing any frequency component in the time-frequency domain.

Once the start point, peak point and end point of p wave, QRS complex, ST segment and T wave are detected from each lead, there is a need to calculate the time interval between every point. Each point in an ECG signal is located with respect to its value of amplitude or voltage in Y-axis and time value in X-axis. Earlier, scaled ECG papers were used to represent the ECG signal, where each unit box represented 40 milliseconds of time on the x-axis and 0.1 mv on the y-axis. During digitalization, the analog signal is converted into digital, which is mapped on to a scaled function similar to that of the scaled ECG paper to identify the points present in the ECG. Further we implemented the Nearest-Neighbor Interpolation method to calculate the time interval between points.

C. Nearest-Neighbor Interpolation

A normal ECG signal composes of P wave, which is less than 0.25mV in amplitude and less than 0.12 sec in width. Q wave is less than 0.04 sec in duration and less than 25% of R wave, where R wave has amplitude of at least 0.5 mV in limb leads and 1mV in pericardial leads. The S wave amplitude is greater than R wave in lead V1 and smaller than R wave in lead V6. T wave does not exceed 0.5 mV in limb leads and it is 1 mV in pericardial leads. Also, normal PR interval ranges between 0.12 and 0.20 sec, QRS duration ranges between 0.07 to 0.10 sec, QT interval ranges between 0.33 to 0.43 sec and ST interval is 0.30 to 0.34 sec. In order to identify the points in the ECG we make use of the Nearest-Neighbor interpolation. By definition, Nearest-Neighbor interpolation is a process of determining approximate values of a function within a range bounded by two function values. In ECG signal analysis, once the points are detected, the interval between these points is determined using the Nearest-Neighbor interpolation method.

In this research work, we adopted a variable threshold method for peak amplitude measurement. We need to measure the amplitude of each segment, particularly R point and the peak of ST segment with J point. Since, we know the amplitude of every point on normal ECG signal, P wave which is less than 0.25mV, Q wave has amplitude of 0.30 -mV, R wave has amplitude of 1.5 mV, S wave has amplitude of 0.40 -mV and T wave does not exceed 0.5 mV. An amplitude threshold is determined as a fraction of the largest positive valued factor of that array. The threshold is fixed by defining d_{th} for each ECG signal, which is calculated using the following equation,

$$d_{th} = \frac{1}{2}((0.5 \times \max(\text{point}[q]) + mean(\text{point}[q])) \quad (1)$$

The peak amplitude measurement is greatly affected by baseline wandering, which is the instability of the base line or change in isoelectric line position, which is likely to be moving up or down. This is primarily caused by improper connections in the ECG leads. Filtering the baseline can reduce wandering, thereby increasing the accuracy of amplitude measurement.

Fig. 2 shows all the features extracted (amplitude and time interval of each wave (P, Q, R, S, T, and U)) as required for classification and block localization in the culprit artery.

P wave threshold is 0.20 mV, Q wave threshold is -0.15 mV, R wave threshold is 0.9 mV, S wave threshold is -0.40 mV and T wave does not exceed 0.5 mV. When the peak is above the threshold, it is detected successfully. The entire feature identification and extraction process is presented in Fig. 2. Classification of ECG signal for locating block in the culprit artery is the next stage and it is presented in the next section.

D. Classification of ECG Signal

Several techniques for detecting blocks in coronary artery using ST segment in ECG signals have been analyzed. At present, medical equipment can vigilant a physician to the probable changes in ST and T wave deviations. But a specialist must still inspect these changes to decide whether they are due to blockage or not. Present methods of automatic classification for location of blocks are time-consuming and cannot discover blocks in real-time. This motivates a need for a capable system for automatic block detection. ECG Signal classification based on Information Fuzzy Network (INF) with rule based criteria distinguishes between different ECG signals that have different characteristics based on the extracted features analyzed earlier in this paper.

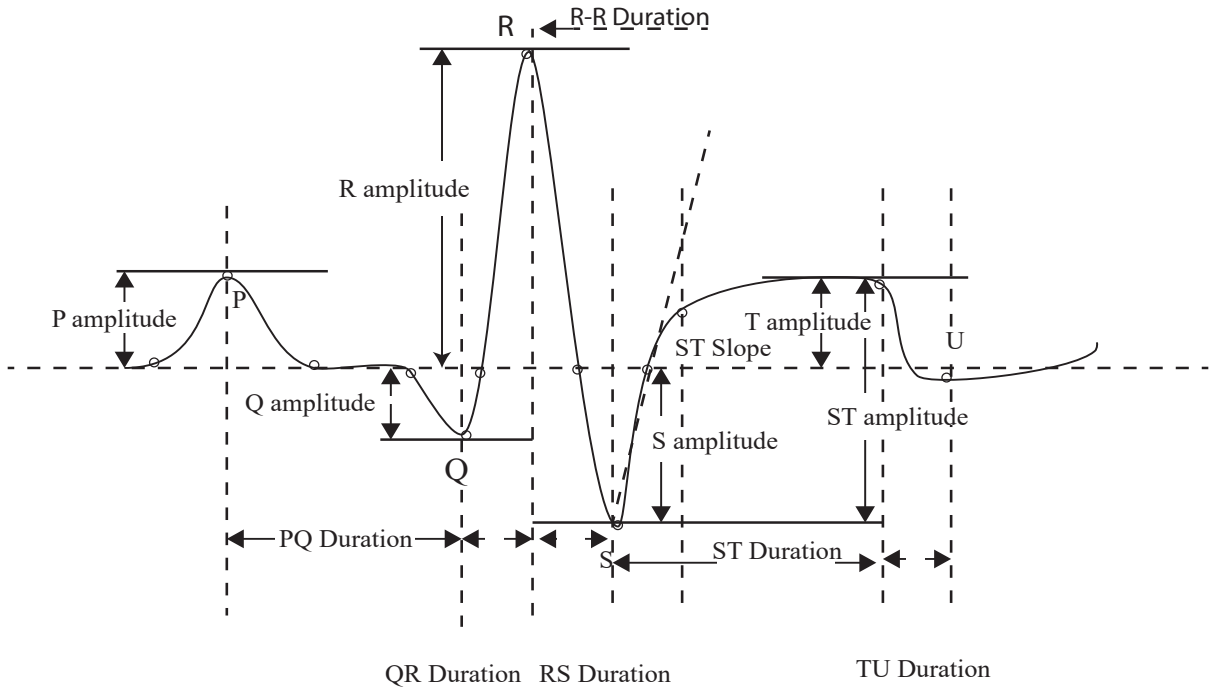


Fig.. 2. Extracted features from one ECG cycle.

1. Differentiating Ischemia and Non-Ischemic ECG Signals

An easy to-use, simple and accurate classification for locating the block has been proposed based on the evaluation of ST-segment changes in a 12- lead ECG that predicts the site of block in patients. There are several techniques employed for detecting Ischemia. However there are two methods which are widely used: coronary angiography and echocardiogram. While angiography is an invasive method for detection with 98% accuracy, echocardiogram is a non-invasive method with an accuracy of around 85%.

A block in the coronary artery causes variations in the depolarization of the heart's ventricular tissue. If the cardiac tissue is Ischemic, it results in improper depolarization. This shows an improper variation in current pulse, which causes a depression in the T-Q segment and an elevation in the ST segment. ST deviation can be calculated by using equations (2), (3) and (4) as follows,

To find ST Depression,

$$PR\,po\,\mathrm{int}(i'\,j) = \frac{1}{2}ceil(R\,loc(i,j) - (S_{off}(i,j) - Q_{on}(i,j)) \tag{2}$$

To find ST Elevation,

$$ST\ \ po\ \mathrm{int}(\,i,\,j) = \frac{1}{2}ceil\ (Tloc\ \ (i,\,j) - T_{off}\ \ (i,\,j) - T_{on}\ (i,\,j)) \tag{3}$$

To find ST Deviation,

$$ST\ Deviation(i,j) = abs(x(PR\ po\,\mathrm{int}(i,j),i) - x(ST\ \ po\,\mathrm{int}(i,j),i)) \tag{4}$$

Where ceil is the Ceiling Function, the least integer greater than or equal to x abs is the Absolute Value, indicating the distance of the point or number from the zero point; the value of T_{on} and S_{off} is the first offset of S signal and T_{on} is the first onset of T wave which separates ST segment. From peak detection, zero crossing point behind the peak is considered as onset and zero crossing point ahead of the peak is detected as offset.

The ST deviation thus obtained is divided by 100; a normal ST deviation would be less than 0.148. Ischemia is detected if the ST Deviation obtained is greater than 1 (these condition are fixed using cardiologist input). We confirmed that the ST deviation is caused by a block in coronary artery and if the condition is false, the ST deviation may be due to Non-ischemic causes. The flowchart given in Fig. 3 illustrates the method of differentiating Ischemia and Non-Ischemia.

2. Rule based Criterions for identifying the Culprit Artery and Localizing the Thrombus

One of the main objectives of this article is to identify the culprit artery from Ischemic ECG signals and finally to locate the exact position of the block as proximal, middle or distal. For this purpose, the classification process depends on the conditions given by [39]. The conditions for predicting the culprit artery is obtained from practical cases in real time situations. However, the main issue with this is the fact that the conditions are derived from independent leads. To overcome this problem, we adopted an ECG lead selector for analyzing the corresponding lead. The major conditions for identifying the culprit artery are thus: if ST elevation in leads L2, L3, aVF (limb lead), and L2 elevations are greater than L3 elevation, it is detected as Left Circumflex Artery (LCx). If L3 elevation is greater than L2 elevation, then it is detected as Right Coronary Artery (RCA). One more condition for RCA is that the V1 lead (Chest lead) must be greater than the V3 lead. Similarly, if V3 elevation is greater than the V1 lead, then it is predicted as the Left Anterior Descending Artery (LAD).

Further identifying the culprit artery, the exact spot of the block needs to be localized. Since identification in LAD is the lengthiest process as compared to the other branches, locating the block in LAD is valuable. Hence LAD is considered for locating the block. The primary condition is the presence of ST deviation in the lead aVF, L2 and L3, it is then predicted as the Proximal to major septal artery LAD. The next condition is the ST elevation in L1, avl, V5, V6 and it is then a proximal to major diagonal artery of LAD. Similarly if ST elevation is in L2, L3 and aVF, then it is distal to major septal artery of LAD. Finally ST elevation in V2, V3, and V4 with Right Bundle Branch Block (RBBB) is predicted as Proximal LAD. The remaining sub conditions for identification and localization are depicted in Fig. 4 and Fig. 5.

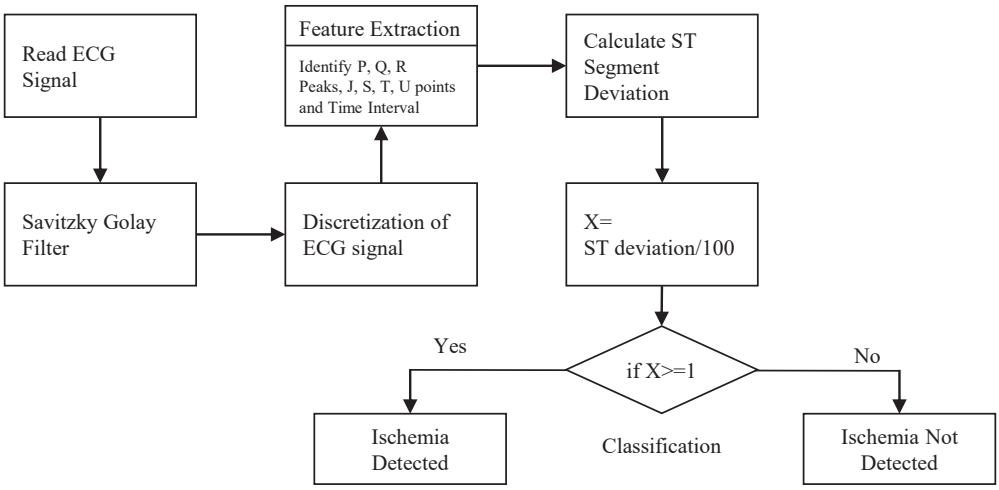


Fig. 3. Flow chart for differentiating Ischemia and Non-Ischemic.

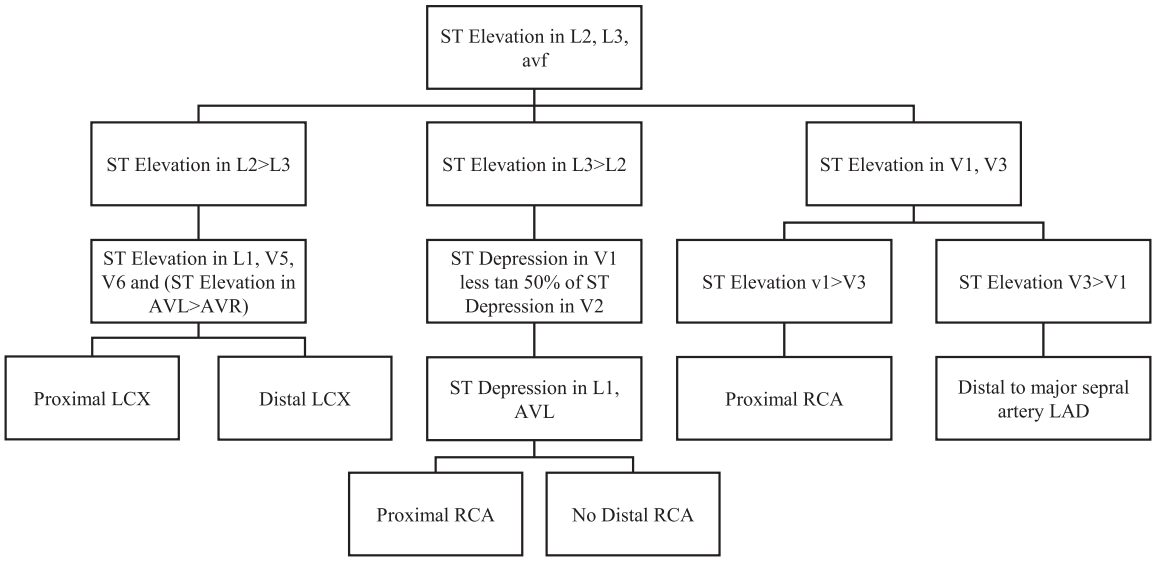


Fig. 4. Flow chart to identify the Culprit Artery.

E. Information Fuzzy Network for ECG Classification and Identification of Block

The proposed approach for ECG classification and block location is based on the Information-Fuzzy Network (IFN). The IFN produces a ranked list of outputs relevant to a given set of input features as a result of inducing a correct classification task. Fig. 6 shows the model of the information fuzzy network.

IFN is a greedy machine-learning algorithm for supervised learning. IFN consists of many nodes. Every node inside the net is called as the inner or hidden node. Each node is weighted with conditional mutual information. IFN uses the conditional information in order to classify features during the identification stage. The set of nodes in the column are called layers, and each layer carries a condition for feature classification. However, there cannot be more than one condition in a layer. The IFN model is typically more stable, which means that small variations in the training set will affect it less, when compared to other models.

The whole classification process is made functional by imposing the previous conditions defined over the IFN. The root node is associated with the features that are extracted during the feature identification and extraction stage (such as amplitude and time interval). Each hidden layer consists of nodes representing equivalence classes of the corresponding input variable. There are three hidden layers other than the root layer. Root layer is considered as the layer 0, while layer 1, layer 2 and layer 3 are the hidden layers and the target layer is the output layer. Layer 1 has the attributes or conditions for the Ischemic or Non-Ischemic causes. Layer 2 consists of two classes as shown in Fig. 6. The first class has the attributes or conditions for the classification of supraventricular from ventricular tachycardia, while the second class has the attributes or conditions for identifying the culprit artery. The IFN automatically determines the equivalence classes as contiguous sub-ranges of input values. Finally, the output layer has five output variables that indicate the location of the block in the culprit artery.

1. Training IFN

A training set is a set of data employed to uncover the potentially predictive relationships between the input and the output. A test set is a set of data used to evaluate the strength and effectiveness of a prognostic association. Test and training sets are used in IFN, so that it can make predictions on new data. The accuracy and reliability of the block detection depends on the type of training set used. The database from MIT Physionet databank is divided into two splits: 30% of data to train and 70% to test the system.

Training phase is also called as weight updating or tuning phase

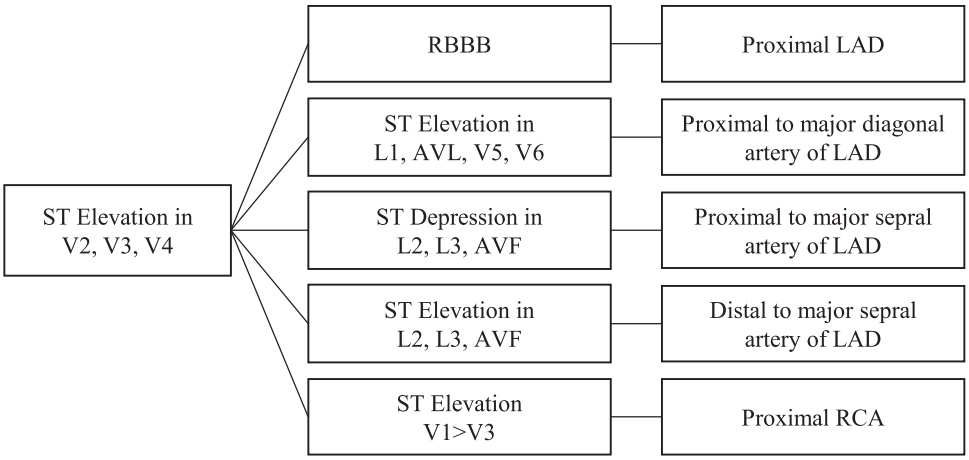


Fig. 5. Flow chart for localizing the Thrombus.

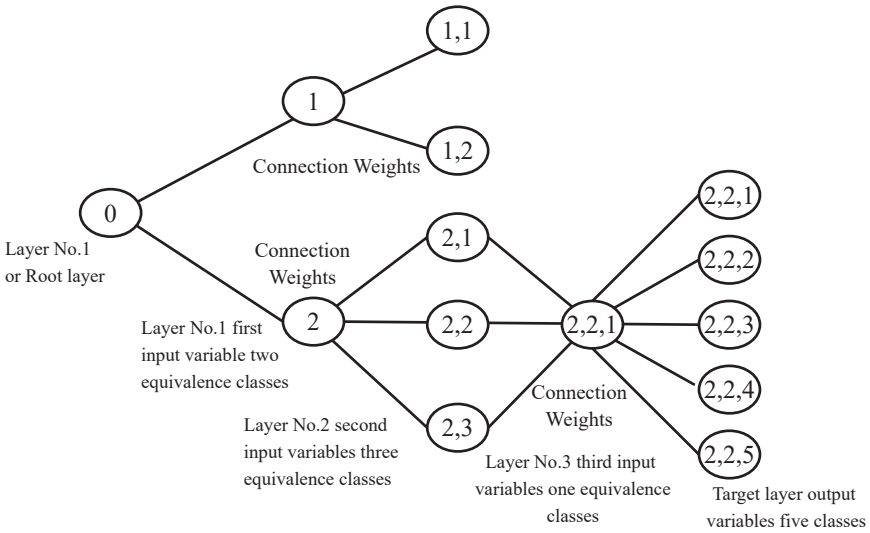


Fig. 6. IFN model for ECG classification and identification of block using Culprit Artery.

of the system. During training, the ECG signal from the database is applied to the system and the corresponding output is analyzed for error function. The objective of the training phase is to reduce the error function so that the result obtained from testing would be more accurate. The training error of the estimated conditional mutual information between an input attribute Ai and the target attribute Bi in training phase is evaluated by using the likelihood-ratio statistic. The likelihood-ratio test is a general-purpose technique for testing the null hypothesis Ho, that two random variables are statistically independent. This likelihood-ratio can be used to compute a p-value or compared to a critical value to decide whether to reject the null model in favor of the alternative model. The default significance level (p-value) used by the IFN algorithm is 0.1 %. Likelihood-ratio statistic is given by,

$$R^2(A_i; B_i / n) = 2 \bullet (In2) \bullet E^*(n) \bullet MI(A_i; B_i / n) \quad (5)$$

In the above equation (5) $E^*(n)$ is the number of cases correlated with the node n. After detecting the training phase error, it needs to be minimized by adjusting the weighted function of IFN. The weights are modified in proportion to their contribution to the error. Updating is done after each cycle using the conditional mutual information as mentioned in the learning section. It must be pointed out that the weights (W) are adjusted to ensure that $-0.1 \le W \le 0.1$ for each revised step, so that the chances of overshooting of the weights may be minimized, in the course of smoothly approaching a minimum error solution.

2. *Testing IFN*

Testing is done by computing the features over a new signal sample and choosing the class label corresponding to the IFN with the highest output. Subsequent to training, the recommended model encodes in its weights information regarding class discrimination in the feature space called label, which is the desired Information Fuzzy Classifier. In the testing phase, the 70% of data is considered as input to the IFN model and its performance is evaluated. During the testing phase, the system labels the test vector in the dimensional space. The output layer yields values by indicating the training pattern in the corresponding output classes. Once the Information Fuzzy Network identifies the input-output relationships, testing consists of generating the test cases that represent the union of combinatorial tests for input variables included in the IFN model at each single output. The results drawn from the proposed methodology are given in the next section.

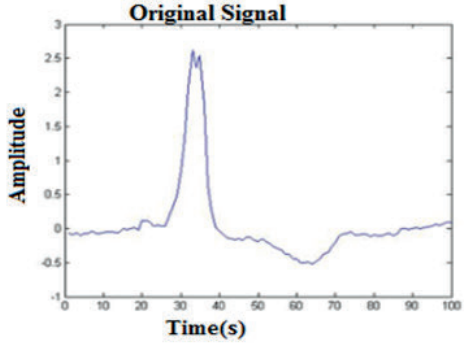
IV. Results

In this section, we present the results using the Information Fuzzy Network approach for automated block detection in the culprit artery. We perform an automated input-output analysis by training the Information Fuzzy Network. The proposed method is implemented using Intel i5 environment over a Personal Computer with 2.99 GHz CPU and 8GB RAM running on Windows 7 OS. The induced information fuzzy models are utilized for block detection. We evaluate

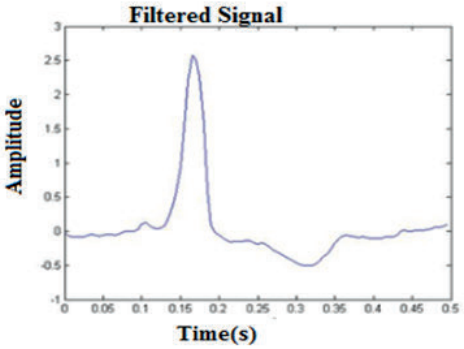
the performance of the proposed method by comparing Sensitivity, Specificity, Positive Predictive Value and Negative Predictive Value and Classification Accuracy as evaluation metrics.

A. ECG Database

In general, classifiers with a higher number of output classes require more features, which imply the need for a larger database. However, to prevent over-training, which would mean great performance on the training set, but a poor performance in testing sets, the database considered represents a well-defined differentiating feature set. Hence, we selectively combined some of the databases including ECG signals of patient with Ischemic Myocardial Infarction, Ventricular Tachycardia, Supraventricular Tachycardia and ECG signal of normal patient. The proposed methodology is implemented using MATLAB R2013a platform, and the experimental results are analyzed and compared with the conventional methods. The data chosen from MIT Physionet databank for training and testing are: Long-Term ST Database, consisting of 86 records that are 21 to 24 hours long, and contain two or three ECG signals, annotated beat-by-beat. With respect to the ST episodes as the rhythm changes, the signal quality changes. Each record includes ST level time series based on 16-second averages centred on each beat. Spontaneous Ventricular Tachyarrhythmia Database contains 135 pairs of RR interval time series, recorded by implanted cardioverter defibrillators in 78 subjects. Each series contains between 986 and 1022 RR intervals. One series of each pair includes a spontaneous episode of Ventricular Tachycardia (VT) or Ventricular Fibrillation (VF) and the other is a sample of the intrinsic (usually sinus) rhythm. T-Wave Alternans Challenge Database contains 100, 2, 3, and 12-lead ECG records sampled at 500 Hz with 16-bit resolution over a ± 32 mV range, including subjects with risk factors of sudden cardiac death as well as healthy controls and synthetic cases with calibrated amounts of T-wave Alternans [37]. The output ECG signals and results are illustrated in Fig. 7(a). Fig. 7(a) shows the original ECG signal from database and Fig. 7 (b) shows the filtered ECG signal using Savitzky-Golay filter.



(a)



(b)

Fig. 7. (a) Original ECG signal (b) Filtered ECG signal using Savitzky-Golay filter.

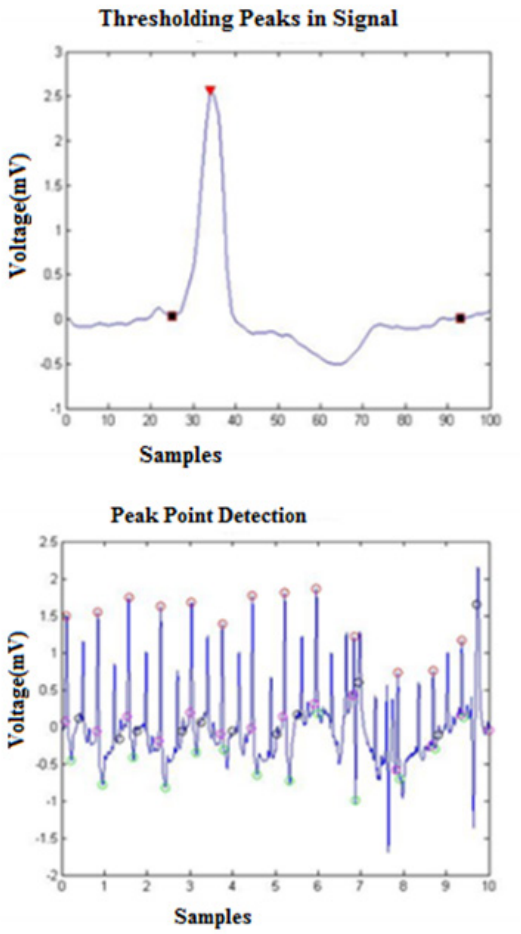


Fig. 8. Detection of points in ECG signal.

The original ECG signal shown in Fig. 7(a) is given as the input signal to the Savitzky-Golay filter. The output from the filter is further used for feature extraction process. The detection of points in the ECG signal is shown in Fig. 8. This figure shows the point detection by Stockwell transform. The extracted features such as time interval and amplitude of each point on ECG are tabulated in table I.

TABLE I. Time Interval And Amplitude Of ECG Segment

ECG Segments	Time Interval (ms)	ECG points	Amplitude (mV)
RS segment	0.06271	S	0.4590
RR segment	0.14988	T	0.4090
ST segment	0.35	Q	0.3490

Table I shows the time interval of RS segment, RR segment and ST segment and the amplitude of the ECG Signal. In order to prove the efficiency and accuracy of the proposed method, the analytical performance of the proposed block detection using IFN approach is validated by the metrics given in the next section.

B. Performance Evaluation

Performance evaluation is carried out by comparing classification accuracy of the proposed methodology with existing classifiers such as Support Vector Machine [40], Compound Neural Network [41] and Rough-Set-Based Inference Engine [42, 43] for ECG classification with five parameters: Sensitivity, Specificity, Positive Predictive Value (PPV), Negative Predictive Value (NPV) and Classification Accuracy.

The five performance parameters described above depicts the performance of the proposed methodology in terms of classification

of ECG signal and the detection of the block. The result shows the effectiveness of the system performance (see Table II). The efficiency of the proposed method can be well shown with the aid of comparative analysis. A comparison of the proposed method with other existing methods is given in the next section.

TABLE II. Performance Evaluation of the Proposed Method

Performance Parameters	Proposed Thrombus Detection using Information Fuzzy Network (IFN)
Sensitivity	87.50%
Specificity	100%
PPV	100%
NPV	83.33%
Classification Accuracy	92.30%

C. Performance Comparisons

In this section, we compare the proposed method with Generalized Least Square-Support Vector Machine method, Compound Neural Network and Rough-Set-Based Inference Engine. In the first method, five types of ECG beats and the features from three transform approaches are independently classified using Least Square-Support Vector Machine (LS-SVM) [42].The second method is based on a Compound Neural Network (CNN) to classify ECGs as normal or carrying an Atrio-Ventricular heart Block (AVB) [41]. This method uses three different feed forward multilayer neural networks. The third method is a rule-based rough-set decision system for the development of a disease inference engine [42], [43]. The proposed system is based on the criteria from popular medical literature and feedback from reputed cardiologists. All the performance evaluation metrics are used for the comparison are tabulated in table III.

TABLE III. Comparison of the Proposed Method with other Existing Methods

Performance Parameters	Generalized Least Square-Support Vector Machine [42]	Compound Neural Network [41]	Rough Set Based Inference Engine [42, 43]	Information Fuzzy Network (IFN) (Proposed Method)
Sensitivity	84.62	65.54	48.00	**87.50**
Specificity	83.33	92.25	96.00	**100**
PPV	81.48	63.47	85.30	**100**
NPV	86.21	92.09	91.70	**83.33**
Classification Accuracy	85.59	87.01	91.00	**92.30**

From Table III, it is clear that the ECG signal can be used as a reliable indicator for detecting heart blocks. In this research work, ECG analysis is effectively used as a non-invasive tool for the classification and location of the block in the coronary artery. Here, our proposed methodology was compared with Compound Neural Network, Rough Set based Interface Engine and Least Square-Support Vector Machine in terms of Sensitivity, Specificity, and Classification Accuracy. The obtained results proved a valid performance improvisation.

The proposed method for block detection in the culprit artery based on traditional conditions derived from real time incidents is presented. These conditions are formed as a generalized algorithm and implemented using Information Fuzzy Network. The results of the proposed method are tabulated in Table II and Table III. The results of the filtering method used are presented in Fig. 7. It shows the original ECG signal from the database and the filtered ECG signal. From Table III, it is clear that the proposed system is highly efficient, as the rate of Sensitivity is 87.5%, Specificity is 100% and Classification Accuracy is 92.30%; which are more accurate as compared to other methods. Also, the rate of Positive Predictive Value is 100% and Negative Predictive Value is 83.33% and classified output is high when compared to other methods. Similarly, to prove the effectiveness of the proposed methodology, it has been compared with other conventional works that use different ECG classification to analyze ECG signal. The comparisons results presented in table 3 depicts that the results achieved by implementing the proposed method are far better than the works proposed earlier.

V. Conclusion

In this research work, a method for identification of culprit artery along with detection of block location based on Information Fuzzy Network using ECG signals is developed. The decision of the proposed classification results are obtained from traditional criteria. The proposed detection system is validated successfully with 264 ECG samples with ST episodes variation, RR interval time series variation and T-Wave Alternant signals from both genders and different ages. The proposed method first distinguishes the ECG signals as Ischemic or Non-Ischemic, identifies the culprit artery and finally locates the block in the identified artery. The proposed detection system achieves 92.30% of accuracy. All the input voltage parameters and the output of the logical expressions related to the proposed detection criteria are expressed in the IFN system.

In future, we are intended to apply deep networks which could further enhance the results of the proposed method. It is also possible to employ various other soft computing techniques to find the percentage of blockage in the culprit artery.

References

[1] R. S. Bruning and M. Sturek, “Benefits of exercise training on coronary blood flow in coronary artery disease patients,” *Progress in cardiovascular diseases,* Vol.57, No.5, 2015, pp.443-53.

[2] A. Arbab-Zadeh and V. Fuster, “The myth of the vulnerable plaque: transitioning from a focus on individual lesions to atherosclerotic disease burden for coronary artery disease risk assessment,” *Journal of the American College of Cardiology,* Vol.6, No.8, 2015, pp.846-855.

[3] P. Guzik and M. Malik, “ECG by mobile technologies,” *Journal of electrocardiology,* Vol.49, No.6, 2016, pp.894-901.

[4] C. Haarmark, R. P. Hansen, E. Vedel-Larsen, S. H. Pedersen, C. Graff, P. Mads Andersen, M.D. Egon Toft, F. Wang, J. J. Struijk, MScEng, K. Jørgen and Kanters, “The prognostic value of the Tpeak-Tend interval in patients undergoing primary percutaneous coronary intervention for ST-segment elevation myocardial infarction,” *Elsevier, Journal of Electrocardiology,* Vol.42, 2009, pp.555- 560.

[5] S. Ansari, N. Farzaneh , M. Duda, K. Horan and H. Andersson, “A Review of Automated Methods for Detection of Myocardial Ischemia and Infarction using Electrocardiogram and Electronic Health Record,” *IEEE Reviews in Biomedical Engineering*, Vol. 10, 2017, pp. 264-297.

[6] G. Stephen Ellis, M. Tendera, A. Mark de Belder, J. Ad van Boven, P. Widimsky, L. Janssens, H.R. Andersen, A. Betriu, S. Savonitto, J. Adamus, Z. Jan Peruga, M. Kosmider, O. Katz, T. Neunteufl, J. Jorgova, M. Dorobantu, L. Grinfeld, P. Armstrong, R. Bruce Brodie, C. Howard Herrmann, G. Montalescot, F.-J. Neumann, B. Mark Effron, S. Elliot, M. Barnathan, and J. E. Topol, “Facilitated PCI in Patients with ST-Elevation Myocardial Infarction,” *The classification England journal of medicine,* Vol.358, No.21, 2008, pp.2205-2217.

[7] O. Fröbert, B. Lagerqvist, K. Göran Olivecrona, E. Omerovic, Thorarinn Gudnason, M. Maeng, M. Aasa, O. Angerås, F. Calais, M. Danielewicz, D. Erlinge, L. Hellsten, U. Jensen, C. Agneta Johansson, A. Kåregren, J. Nilsson, L. Robertson, L. Sandhall, I. Sjögren, O. Östlund, J. Harnek, and K. Stefan James, “Thrombus Aspiration during ST-Segment Elevation Myocardial Infarction,” *The classification England Journal of Medicine,* Vol.369, No.17, 2013, pp.001–011.

[8] C.K. Roopa and B.S. Harish, “An Empirical Evaluation of Savitzky-Golay

(SG) Filters for Denoising ST Segment," *Springer, Cognitive Computing and Information Processing,* Vol.801, 2018, pp.18-28.

[9] D. Da Costa, W.J. Brady, and J. Edhouse. "Abc of clinical electrocardiography: Bradycardias and atrioventricular conduction block," *BMJ: British Medical Journal,* Vol. 324, No.7336, 2002, pp. 535-538.

[10] S. Padhy, L. Sharma, and S. Dandapat, "Multilead ECG data compression using SVD in multi resolution domain," *Biomedical Signal Process Control,* vol. 23, 2016, pp. 10–18.

[11] S. Padhy and S. Dandapat, "Third-order tensor based analysis of multilead ECG for classification of myocardial infarction," *Biomedical Signal Process Control,* vol. 31, 2017, pp. 71–78.

[12] J. Garcıa, M. Astrom, J. Mendive, P. Laguna, and L. Sornmo, "ECG based detection of body position changes in ischemia monitoring," IEEE Transactions on Biomedical Engineering, vol. 50, no. 6, pp. 677–685, 2003.

[13] M. Schmidt, M. Baumert, A. Porta, H. Malberg, and S. Zaunseder, "Two dimensional warping for one-dimensional signals–conceptual frame work and application to ECG processing," *IEEE Transactions Signal Process,* vol. 62, no. 21, 2014, pp. 5577–5588.

[14] M. Hadjem, F. Nait-Abdesselam, and A. Khokhar, "ST-segment and T wave anomalies prediction in an ECG data using RUSBoost," *IEEE 18th Internatinal Conference on e-Health Networking, Application and services,* 2016, pp. 1–6.

[15] H.N. Murthy and M. Meenakshi, "ANN, SVM and KNN classifiers for prognosis of cardiac ischemia—A comparison," *Bonfring International Journal of Research Communication Engineering,* vol.5, no. 2, 2015, pp. 7–11.

[16] A.K. Bhoi, K.S. Sherpa, and B. Khandelwal, "Classification probability analysis for arrhythmia and ischemia using frequency domain features of QRS complex," *International Journal of BioAutomation,* vol. 19, no. 4, 2015, pp. 531–542.

[17] S.S. Nidhyananthan, S. Saranya, and R.S.S. Kumari, "Myocardial infarction detection and heart patient identity verification," *International Conference Wireless Communication Signal Processing Network,* 2016, pp. 1107–1111.

[18] L.N. Sharma, R.K. Tripathy, and S. Dandapat, "Multiscale energy and eigenspace approach to detection and localization of myocardial infarction," *IEEE Transaction Biomedical Engineering,* vol. 62, no. 7, 2015, pp. 1827–1837.

[19] H. Pereira and N. Daimiwal, "Analysis of features for myocardial infarction and healthy patients based on wavelet," *Conference on Advances in Signal Processing,* 2016, pp. 164–169.

[20] J. Garcıa, P. Lander, L. Sornmo, S. Olmos, G. Wagner, and P. Laguna, "Comparative study of local and Karhunen–Loeve-based STT indexes in recordings from human subjects with induced myocardial ischemia," *Computer And Biomedical Research,* vol. 31, no. 4, 1998, pp. 271–292.

[21] Y.L. Tseng, K.S. Lin, and F.S. Jaw, "Comparison of support-vector machine and sparse representation using a modified rule-based method for automated myocardial ischemia detection," *Computation and Mathematical Methods in Medicine*, vol. 2016, 2016, pp. 1–8.

[22] F. Jager, G.B. Moody, and R.G. Mark, "Detection of transient ST segment episodes during ambulatory ECG monitoring," *Computers and Biomedical Research,* vol. 31, no. 5, 1998, pp. 305–322.

[23] T.P. Exarchos, C. Papaloukas, D.I. Fotiadis, and L.K. Michalis, "An association rule mining-based methodology for automated detection of ischemic ECG beats," *IEEE Transaction Biomedical Engineering,* vol. 53, no. 8, 2006, pp. 1531–1540.

[24] J. Park, W. Pedrycz, and M. Jeon, "Ischemia episode detection in ECG using kernel density estimation, support vector machine and feature selection," *Biomedical Engineering Online,* vol. 11, no. 1, 2012.

[25] J.T.Y. Weng, J.J. Lin, Y.C. Chen, and P.C. Chang, "Myocardial infarction classification by morphological feature extraction from big 12-lead ECG data," *Trends and Applications in Knowledge Discovery and Data Mining,* vol. 8643.

[26] N. Safdarian, N. J. Dabanloo, and G. Attarodi, "A new pattern recognition method for detection and localization of myocardial infarction using T wave integral and total integral as extracted features from one cycle of ECG signal," *Scientific Research Publishing,* vol. 7, 2014, pp. 818–824.

[27] P. Kora and S. R. Kalva, "Improved Bat algorithm for the detection of myocardial infarction," *Springer Plus,* vol. 4, no. 1, 2015.

[28] C. C. Lin,W. Hu, and Y.W. Lin, "A wavelet-based high-frequency analysis of fragmented QRScomplexes in patients with myocardial infarction," *Computing in Cardiology Conference,* 2015, pp. 565–568.

[29] S. Banerjee and M. Mitra, "Application of cross wavelet transform for ECG pattern analysis and classification," *IEEE Transaction Instrumentation and Measurment,* vol. 63, no. 2, 2014, pp. 326–333.

[30] U.R. Acharya, H. Fujita, V.K.Sudrashan, S.L. Oh, M. Adam, E .W. Joel Koh, J.H. Tan, D.N Ghista, R.J. Martis, C.K. Chua, C.K. Poo and R.S. Tan, "Automated detection and localization of myocardial infarction using electrocardiogram: A comparative study of different leads," *Knowledge Based System,* vol. 99, 2016, pp. 146–156.

[31] U. R. Acharya, H. Fujita, V.K. Sudrashan, S. L. Oh, M. Adam, E .W. Joel Koh, J. H. Tan, D.N Ghista, R.J. Martis, C.K. Chua, C.K. Poo and R.S. Tan , "Automated characterization and classification of coronary artery disease and myocardial infarction by decomposition of ECG signals: A comparative study," *Information Science,* vol. 377, 2017, pp. 17–29.

[32] B. Liu, J. Liu, G. Wang, K. Huang, F. Li, Y. Zheng, Y. Luo and F. Zhou, "A novel electrocardiogram parameterization algorithm and its application in myocardial infarction detection," *Computers in Biology and Medicine,* vol. 61, 2015, pp. 178–184.

[33] C.A. Bustamante, S.I. Duque, A. Orozco-Duque, and J. Bustamante, "ECG delineation and ischemic ST-segment detection based in wavelet transform and support vector machines," *Pan American Health Care Exchanges*, 2013, pp. 1–7.

[34] N. A. Bhaskar, "Performance analysis of support vector machine and neural networks in detection of myocardial infarction," *Procedia Computer Science,* vol. 46, 2015, pp. 20–30.

[35] E. A. Maharaj and A. M. Alonso, "Discriminant analysis of multivariate time series: Application to diagnosis based on ECG signals," *Computational Statistics Data Analysis,* vol. 70, 2014, pp. 67–87.

[36] G. Bortolan and I. Christov, "Myocardial infarction and ischemia characterization from T-loop morphology in VCG," *Computers in Cardiology,* 2001, pp. 633–636.

[37] B. Biswal, "ECG signal analysis using modified S-transform", *Healthcare Technology Letters,* Vol. 4, 2017, pp. 68–72.

[38] N. Kannathal, U. Rajendra Acharya, P. Joseph, L. C. Min and J. S. Suri, "Analysis of Electrocardiograms", *Advances in Cardiac Signal Processing,* Vol. 22, 2007, pp. 55-82.

[39] I. Tierala, C. Kjell Nikus, S. Sclarovsky, M. Syvänne, M. Eskola, "Predicting the culprit artery in acute ST-elevation myocardial infarction and introducing a new algorithm to predict infarct-related artery in inferior ST-elevation myocardial infarction: correlation with coronary anatomy in the HAAMU Trial", *Journal of Electrocardiology,* Vol. 42, 2009, pp. 120–127.

[40] D. Tomar and S. Agarwal, "Feature Selection based Least Square Twin Support Vector Machine for Diagnosis of Heart Disease", *International Journal of Bio-Science and Bio-Technology,* Vol.6, No.2, 2014, pp. 69-82.

[41] N. Kumar Dewangan and S.P. Shukla, "ECG Arrhythmia Classification using Discrete Wavelet Transform and Artificial Neural Network," *IEEE International Conference On Recent Trends In Electronics Information Communication Technology,* 2016, pp. 1946-1950.

[42] A. Ratnaparkhi and R. Ghongade, "A Frame Work for Analysis and Optimization of Multiclass ECG Classifier Based on Rough Set Theory," *IEEE International Conference on Advances in Computing, Communications and Informatics,* 2014, pp. 2740-2744.

[43] Chang-Sik Son, Yoon-Nyun Kim, Hyung-Seop Kim, Hyoung-Seob Park and Min-Soo Kim, "Decision-making model for early diagnosis of congestive heart failure using rough set and decision tree approaches," *Journal of Biomedical Informatics,* Vol. 45, 2012, pp. 999-1008.

[44] D. A. Orrego, M.A. Becerra, and E. Delgado-Trejos, "Dimensionality reduction based on fuzzy rough sets oriented to ischemia detection," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* 2012, pp. 5282–5285.

[45] A. Kumar and M. Singh, "Ischemia detection using isoelectric energy function," *Computers in Biology and Medicine,* Vol. 68, 2016, pp. 76–83.

[46] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern classification," *NewYork, NY, USA: Wiley,* 2012.

[47] C. K. Roopa and B. S. Harish, "A Survey on various Machine Learning Approaches for ECG Analysis," *International Journal of Computer Applications,* Vol. 163, 2017.

C. K. Roopa

She received her B.E degree in Information Science and Engineering and M.Tech degree in Computer Engineering from Visvesvaraya Technological University, Belagavi, Karnataka, India. She is currently working as a Assistant Professor at JSS Science & Technology University. She is also pursuing her Ph.D degree in Computer Science from University of Mysore, Karnataka, India. Her area of research includes Machine Learning and Medical Informatics.

B. S. Harish

He obtained his Ph.D. in Computer Science (2011) from University of Mysore. Presently he is working as an Associate Professor in the Department of Information Science & Engineering, JSS Science & Technology University, Mysuru. He was a Visiting Researcher at DIBRIS - Department of Informatics, Bio Engineering, Robotics and System Engineering, University of Genova, Italy. He has invited as a resource person to deliver various technical talks on Data Mining, Image Processing, Pattern Recognition, Soft Computing. He is also serving and served as a reviewer for National, International Conferences and Journals. He has published articles in more than 60 International reputed peer reviewed journals and conferences proceedings. He successfully executed AICTE-RPS project which was sanctioned by AICTE, Government of India. His area of interest includes Machine Learning, Text Mining and Computational Intelligence.

Fuzzy C-Means Clustering with Histogram based Cluster Selection for Skin Lesion Segmentation using Non-Dermoscopic Images

Salam Shuleenda Devi*, Ngangbam Herojit Singh, Rabul Hussain Laskar

Department of Electronics and Communication Engineering, National Institute of Technology Meghalaya (India)

Received 14 June 2019 | Accepted 28 August 2019 | Published 20 January 2020

Abstract

Purpose – Pre-screening of skin lesion for malignancy is highly demanded as melanoma being a life-threatening skin cancer due to unpaired DNA damage. In this paper, lesion segmentation based on Fuzzy C-Means clustering using non-dermoscopic images has been proposed.

Design/methodology/approach – The proposed methodology consists of automatic cluster selection for FCM using the histogram property. The system used the local maxima along with Euclidean distance to detect the binomial distribution property of the image histogram, to segment the melanoma from normal skin. As the Value channel of HSV color image provides better and distinct histogram distribution based on the entropy, it has been used for segmentation purpose.

Findings – The proposed system can effectively segment the lesion region from the normal skin. The system provides a segmentation accuracy of 95.69 % and the comparative analysis has been performed with various segmentation methods. From the analysis, it has been observed that the proposed system can effectively segment the lesion region from normal skin automatically.

Originality/Value – This paper suggests a new approach for skin lesion segmentation based on FCM with automatic cluster selection. Here, different color channel has also been analyzed using entropy to select the better channel for segmentation. In future, the classification of melanoma from benign naevi can be performed.

Keywords

Medical Image Segmentation, Melanoma, Histogram, Fuzzy C-means, Local Maxima, Non-Dermoscopic Image, Cluster Size.

DOI: 10.9781/ijimai.2020.01.001

I. Introduction

Melanoma is one of the threatening skin cancer diagnostics in the human body. It is mainly developed due to unpaired DNA damage by ultraviolet radiation from sunlight. Also triggers mutations which cause the rapid multiplication of skin cells and form malignant tumors. The death rate due to melanoma skin cancer is rapidly increasing as compared to other cancer except lung cancer. In United States, 1-78-560 cases are reported as melanoma in 2018. Out of those, 87-290 cases are noninvasive and 91-270 cases are invasive. Melanoma is curable in early stage, but if it is not, it becomes fatal and hard to treat. About 9-320 people died due to melanoma cancer in the US (men: 5990, women: 3330).

There are two possible ways to lower the death rate from melanoma cancer, i.e. to diminish the number of new tumors by identifying and eliminating or preventing the development of melanoma. Many researchers have developed many systems to identify and prevent the situations but it does not hold great.

Classification of melanoma from benign melanocytic naevi is not an easy task. Hence, the importance of digital image analysis techniques is increasing [1]. The analysis can be done using two different images, i.e. dermoscopic (microscopic) and non-dermoscopic (clinical images). Non dermoscopic images are easily available and accessible as compared to dermoscopic. These images are captured by using conventional user grade cameras which is easily accessible by naked eye [2]. Non-dermoscopic images are preferred over dermoscopic images as they have the benefit of being easily accessible. Whereas dermoscopic images are captures by using a special instrument called dermatoscope which is not easily available. In US, dermoscopic images are used by 50% of the dermatologists for cancer examination. Non-dermoscopic image has also some issues such as non-uniform illumination and less contrast results in tough lesion segmentation.

* Corresponding author.
E-mail address: shuleenda26@gmail.com

II. Literature Review

Image segmentation is much used in the medical field for computer aid detection of different types of cancer as the breast one [3] or the skin one, targeted in this research. Various computerized techniques for skin lesions segmentation are developed by researchers. Different segmentation approaches such as region merging, active contours and thresholding have been used for this purpose [4] [5]. Friedman *et al.*[6], proposed a diagnostic technique using the criteria such as Asymmetry, Border irregularity, Color variation and a Diameter greater than 6

mm for melanoma skin cancer (ABCD acronym). Rosado *et al.* [7], performed a comparative study between the diagnostic results of computer diagnostic systems and trained clinical expert. Here, higher sensitivity is observed with computer systems, but lower specificity (i.e. more false positive than humans). Unsupervised based pigmented skin border detection using statistical region merging algorithm has been proposed. The method comprises of black frame removal, image smoothing and segmentation skin lesion based on region merging [8]. P. G. Cavalcanti et al., [9] utilized a discriminating 3-channel space obtained by principle component analysis to segment the skin lesion. Sadri *et. al*., proposed a system based on wavelet network for dermoscopic images. Here, fixed-grid network is formed without training. Further, network weights calculation and network optimization is done by using orthogonal least squares algorithm. Here, experimental analysis was done using 30 dermoscopic images [10]. An automated lesion segmentation framework has been proposed which consists of a multi-stage illumination correction and texture-based segmentation [11] [12]. High level intuitive features were also proposed to model ABCD criteria used for melanoma detection. It is mainly designed to overcome the issues related with low-level features [13]. Smartphone based melanoma detection has also been proposed. Here, a set of features were extracted for better assessment of the ABCD rule [14]. For non- dermoscopic images, red color channel image along with otsu thresholding has been used to segment the lesion region [15]. Further, principal component analysis has been applied on RGB color space to discriminate the lesion region [16] [17]. Convolutional neural network based lesion region extraction has also been discussed [18]. Skin lesion segmentation is still one of the challenging tasks due to various issues such as scarcity of the image database [19], non-uniform illumination, system performance varying for different images.

Various approaches such as supervised machine learning based segmentation [18], unsupervised approaches (Fuzzy C-means, PCA, etc) have been introduced to overcome the lesion segmentation issues. Recently, Fuzzy C-means (FCM) clustering has been used for medical image segmentation. Ali *et al*. [20] proposed a fuzzy c-means clustering along with mathematical morphology for melanoma segmentation. The aim of the paper is to develop unsupervised skin lesion segmentation for non-dermoscopic image. In FCM, one of the problems is that numbers of cluster should be predefined. To develop an automatic segmentation system based on fuzzy c-means clustering, an automatic system for number of cluster selection is required. In this paper, we develop an automatic skin lesion segmentation system using fuzzy c-means clustering along with histogram property. Histogram property helps to select the number of cluster automatically. The contributions of the paper are:

1. Number of clusters for FCM has been selected automatically using the histogram property. Histogram has been analysed to detect the distribution model (i.e. binomial or normal). First, the two most prominent local maxima of the number of pixel counts are evaluated and the Euclidean distance between them is evaluated. If the distance between them is greater than some threshold value, then the histogram shows normal distribution. Here, threshold value is set by analysis of the histogram of three different cases of skin, i.e. normal skin, melanoma skin and benign naevus skin.
2. The entropy of the different color channels (Hue, Saturation, Value) has been evaluated to choose the better color channel for Fuzzy c-means clustering.
3. Fuzzy c-means has been applied on value channel image to segment the skin lesion. Further, morphological filtering has been done to remove the unwanted artefacts which have been segmented as the lesion region.
4. Further, a comparative analysis has been performed at different levels such as qualitative as well as quantitative.

III. Proposed Method

The complete architecture of melanoma segmentation consists of five stages i.e. Color space conversion, channel image extraction, number of cluster selection, contrast enhancement and segmentation as shown in Fig. 1.

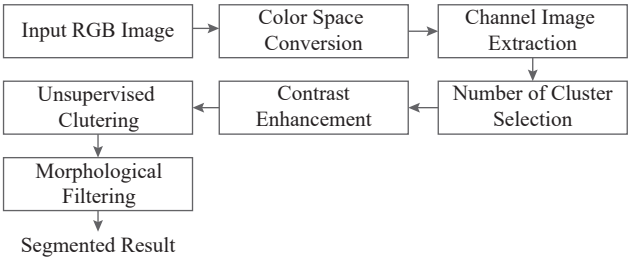


Fig. 1. Proposed method.

A. Image Database

For the experimental analysis, the image dataset of the Department of Dermatology of the University Medical Center Groningen (UMCG) has been used [21] [22]. The images were captured using a Nikon D3 or Nikon D1x body and a Nikkor 2.8/105 mm micro lens maintaining a distance of 33 cm between the lens and the lesion. A total of 170 images (melanoma: 70, naevus: 100) have been used for the experimental purpose. Some of the sample images of the dataset of UMCG are shown in Fig. 2. The sample contains various melanoma cases.

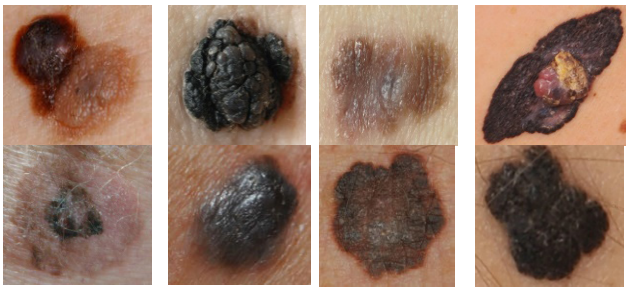


Fig. 2. Sample melanoma images of UMCG dataset.

B. Color Space Conversion

HSV (Hue, Saturation, Value) defines color using familiar comparisons such as color, vibrancy and brightness, with value channel being similar to grayscale image [23]. It is preferred over the RGB color model as it has tendency to perceive color similar to the human eyes. Hue (H), Saturation (S), and Value (V) channel can be defined as

$$H = 60^0 \times H' \tag{1}$$

where $$H' = \begin{cases} undefined & if \quad c = 0 \\ \frac{G-B}{c} \bmod 6, & if \quad m_2 = R \\ \frac{B-R}{c} + 2, & if \quad m_2 = G \\ \frac{R-G}{c} + 4, & if \quad m_2 = B \end{cases}$$

$$V = m_2 \tag{2}$$

$$S = \begin{cases} 0, & if \quad V = 0 \\ \frac{c}{V}, & otherwise \end{cases} \tag{3}$$

where $m_1 = \min(R,G,B)$ is minimum component value, $m_2 = \max(R,G,B)$ is maximum component value and $c = m_2 - m_1$ is chroma component value [15] [17].

C. Channel Image Extraction

One of the major issues of lesion segmentation is the poor contrast between the normal skin and lesion region. Therefore, it is required to examine the contrast in different color channels to select the better one. Here, the color channel selection for segmentation is done by entropy [23] [25] . Entropy provides the measure of information content in images. The entropy results of various color channels (i.e. hue, saturation and value) of some sample images are listed in Table I. More entropy indicates the presence of more information which is defined in eq. (4).

$$E = -\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} P_{ij} log_2(P_{ij}) \quad (4)$$

where, P_{ij} represents the probability density function of the image of gray level (i, j) and M represents the total number of gray levels.

TABLE I. Entropy Results of Different Channel Images

Sample Images	Entropy		
	Hue	Saturation	Value
1	3.0687	6.2543	6.5193
2	5.0831	6.3505	7.3585
3	5.1328	7.0680	7.6415
4	4.3886	6.6313	6.7071
5	3.7526	6.6496	6.7389
6	4.9115	6.5971	6.6750
7	3.2871	6.1576	6.1019
8	3.2119	6.3167	6.5886
9	3.9622	6.3803	6.6492
10	4.3275	6.4314	6.7713

D. Number of Cluster Selection

In Fuzzy c-means, data clustering is done based on the number of clusters given in advance. Here, the number of clusters is selected automatically using histogram of color channel image. The algorithm used for numbers of cluster selection is shown below:

1. Algorithm for Histogram based Cluster Selection:

Step1: Find the most two prominent local maxima of the histogram

Step2: Calculate the distance (dist) between two prominent local maxima

Step3: if dist <= threshold

Step 4: then,

hist=Binomial distribution, Number of Cluster=2, Perform FCM clustering

Out=Segmented Result

else,

hist= normal distribution

Out=Normal Skin

The threshold value is selected by calculating the Euclidean distance between the two prominent local maxima of the histogram of value channel image. Experimental analysis has been conducted to select the better threshold to separate the normal skin and melanoma skin. From the analysis, the threshold value is set as $Th = 5671\ no.of\ pixels$. If the Euclidean distance between two prominent local maxima is greater than the *Th* value, then the image will be considered as normal skin image. If less than *Th*, then the non-dermoscopic image contain melanoma lesion.

E. Contrast Enhancement

Contrast enhancement is used to map the intensity of an image to a specific range. It helps to improve the quality of the image. The contrast of an image can be decrease or increase depending upon the range of the data [22] [24].

F. Fuzzy C-Means Clustering

An unsupervised Fuzzy C-Means (FCM) clustering has been extensively used for medical imaging systems mainly for segmentation [26] [27] [28]. Here, FCM has been used to segment the melanoma skin from non-dermoscopic images. FCM algorithm consists of two different activities. In the first activity, the number of cluster should be provided. In the second activity, data clustering is performed by iteratively searching for a set of fuzzy clusters. The number of cluster selection is done automatically based on the histogram analysis. A membership value v_{ij} is used to indicate the membership degree of the *ith* data point to the *jth* cluster. The main aim of FCM is to minimize the cost function J as

$$J = \sum_{i=1}^{M}\sum_{j=1}^{C} v_{ij}^{n} \left\|x_i - c_j\right\|^2 \text{ where, } n \in [1,\infty] \text{ subject to the constraints,} \quad (5)$$

$$\begin{cases} \sum_{i=1}^{C} v_{ij} = 1, & where\ 0 \le v_{ij} \le 1 \\ \sum_{j=1}^{m} v_{ij} > 0, & for\ i = 1,2, \dots \dots C \end{cases} \quad (6)$$

where, m is the number of data points, $\left\|x_i - c_j\right\|$ is the Euclidean distance between the data point and cluster center. C is the number of clusters ($C \le m$). The membership function v_{ij} and centroids v_i are updated until minimum J is acquired. The algorithm of FCM algorithm is shown below:

Step 1: Select the cluster centers randomly

Step 2: Calculate the fuzzy membership v_{ij}

Step 3: Compute the fuzzy centers

Step 4: Repeat step 2 and 3 until the minimum objective function is obtained

G. Morphological Filtering

Morphological filtering is a non-linear operation which is related to morphological features in an image. It helps to remove as well as recover the unwanted portion of the image segmented as foreground or background in binary image. In binary image, morphological operation creates a new binary image in which the pixel has a non-zero value. Some of the basic morphological operations are erosion, dilation, opening and closing. Here, morphological opening is used to recover lesion region segmented as background [23].

IV. Experimental Results and Discussion

Experimental analysis for lesion segmentation has been done for both quantitative and qualitative analysis. The analysis has been performed using the UMCG dataset which is publicly available. The image dataset consists of melanoma (70 images) and benign naevus skin (100 images). The proposed system has been implemented in MATLAM R2017b, on a computer with Intel Core i3 processor, and 4GB RAM. To reduce the runtime for segmentation, the image has been resized into 250x250 pixels. The proposed system took around 1.7s to segment the lesion region from image having 250x250 pixels. The proposed system consists of color space conversion, channel image extraction, number of cluster selection, contrast enhancement, unsupervised clustering and morphological filtering. Here, RGB image has been converted into HSV image as the HSV color space has capability to perceive the color similar to human eye. From HSV color space, the most preferable color space has been chosen based on the entropy of the various color spaces. Value channel has higher entropy in comparison to other color channels, so it has been used for lesion

segmentation. Further, number of clusters has been selected based on the histogram property as shown in Fig. 3.

From the analysis of the histogram of the various skin conditions, it has been observed that the distance between the most two prominent local maximum of the histogram is less in the skin having lesion. In case of normal skin, the distance between the local maxima is higher in comparison to lesion skin. The threshold value has been used to decide whether the histogram is binomial or not. This threshold value has been set based on the analysis of the histogram of various skin conditions. Moreover, the non-uniform illumination and low contrast of the non-dermoscopic image has been corrected by using contrast enhancement technique. The preprocessed image is segmented by using FCM with automatically selected clusters number to segment the lesion from normal skin. The performance of the proposed method is evaluated by metrics such as sensitivity, specificity and accuracy. The segmentation has been done as a classification problem. Here, the pixels in the images are classified as lesion or normal skin pixels.

The ground truth of the segmented lesion region has been created as shown in Fig. 4 left column. Fig. 4 shows the comparative analysis of the qualitative measure of various segmentation results. The segmentation result of the proposed method has been compared with Otsu_R method [15] in both qualitative and quantitative case. From

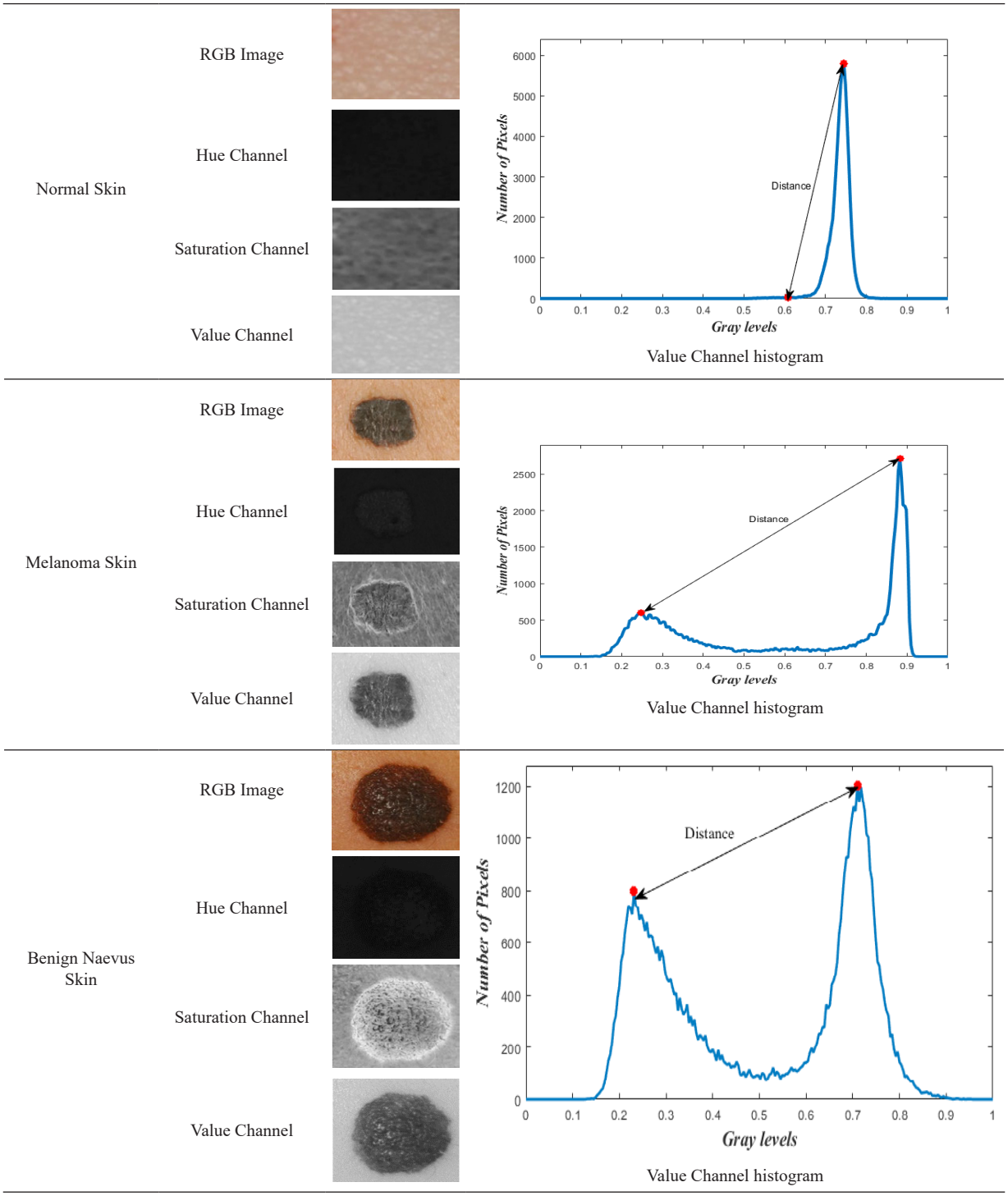


Fig. 3. Details of the histogram for various skin condition.

Table II, it has been observed that the proposed method provides a sensitivity of 90.02%, specificity of 99.15% and accuracy of 95.69%. Our proposed method outperforms the existing method with sensitivity by 11.3%, specificity by 12.78% and accuracy by 12.18%.

Table II. Comparative Analysis Of Various Techniques Using Umcg Dataset

Segmentation Techniques	Image type	Performance statistics (%)		
		Sensitivity	Specificity	Accuracy
Otsu_R [15]	Non-dermoscopic	78.90	86.37	83.51
Proposed method	Non-dermoscopic	90.02	99.15	95.69

V. Conclusion and Future Work

Computer assisted skin lesions segmentation using non-dermoscopic images is of great importance. Due to various factors such as non-uniform illumination and low contrast in non-dermoscopic images, skin lesion segmentation becomes a challenging task. In our proposed method, the information contained in different color channel images is analyzed using entropy. From the analysis, value channel image provides the better results. So, value channel image has been further preprocessed using contrast enhancement to segment the lesion region. Further, pixels clustering has been done using FCM. For FCM, the number of clusters has been selected using local maxima and Euclidean distance property to detect the histogram distribution. Experimental results show that the proposed segmentation method outperforms existing methods, providing an accuracy of 95.69% and sensitivity of 90.02%. In future, lesion classification based on various supervised classifier can be performed. The importance of feature extraction can also be analyzed. The proposed system is mainly designed for non-dermoscopic images. As the images (i.e.

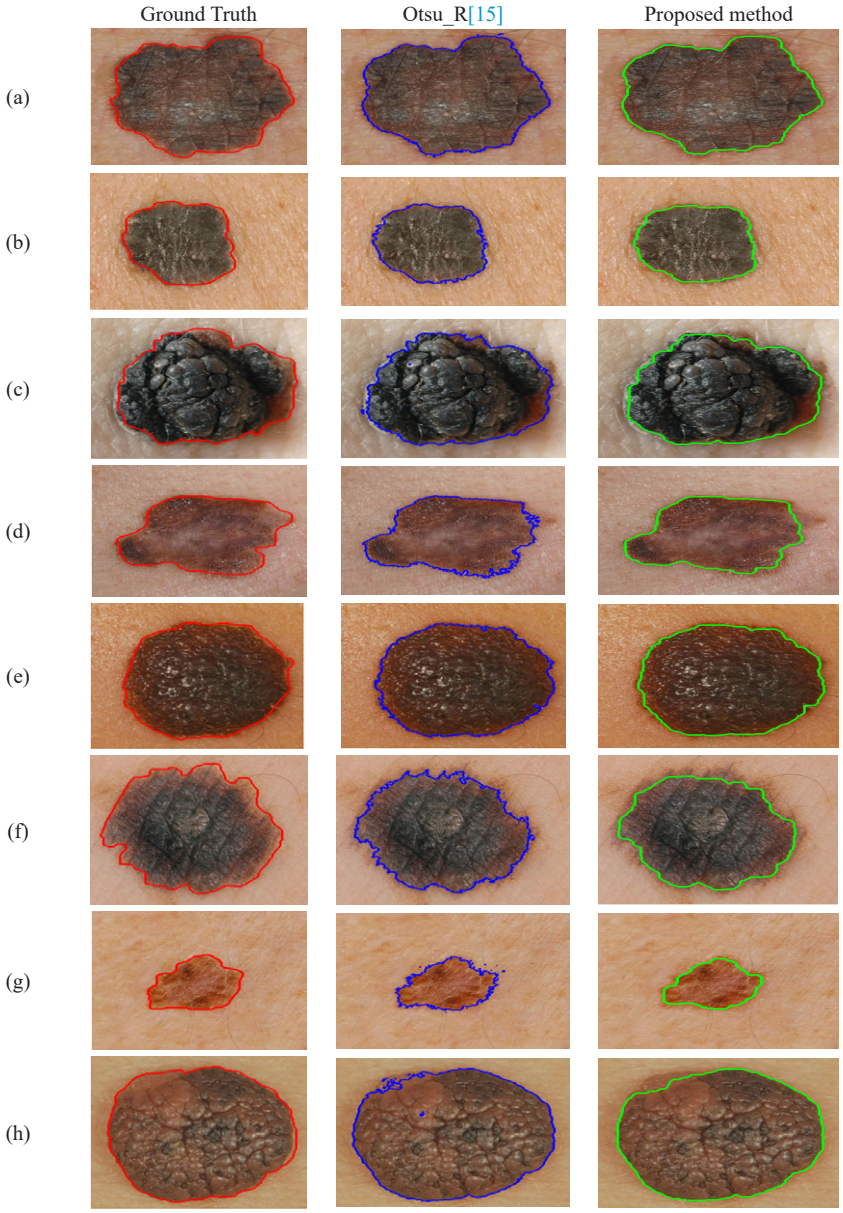


Fig. 4. Segmentation results of skin lesion. ***left column***: input images with ground truth (red line); ***middle column***: segmentation results of otsu_R [15]; ***right column***: segmentation results of proposed method.

non-dermoscopic and dermoscopic) acquired different properties, the proposed system may not be applicable for dermoscopic images, in the present form. With certain customization, the system may also be used for lesion segmentation in dermoscopic images.

References

[1] American Cancer Society: Cancer facts & figures 2016, Atlanta [online], Available: https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/cancer-facts-figures-2019.html

[2] H. C. Engasser and E. M. Warshaw, "Dermatoscopy use by US dermatologists: a cross-sectional survey," J. Amer. Academy Dermatol., 63(3), pp. 412–419, 2010.

[3] L. Belkhodja, D. Hamdadou, "IMCAD: Computer Aided System for Breast Masses Detection based on Immune Recognition," *Int. J. Interactive Multimedia and Artificial Intelligence,* vol. 5(5), pp. 97-108, 2019.

[4] M. Celebi et. al., "Border detection in dermoscopy images using statistical region merging," *Skin Res. Technol*, vol. 14, pp.347–353, 2008.

[5] M. Celebi, H. Iyatomi, G. Schaefer, W. Stoecker, "Lesion border detection in dermoscopy images," *Comput Med Imaging Graph*, vol. 33, pp.148–153, 2009.

[6] R. J. Friedman, D. S. Rigel, and A. W. Kopf, " Early detection of malignant melanoma: The role of physician examination and self-examination of the skin," *CA: A Cancer J. Clinicians*, vol. 35, pp. 130–151, 1985.

[7] B. Rosado, S. Menzies, A. Harbauer, H. Pehamberger, K. Wolff, M. Binder, "Accuracy of computer diagnosis of melanoma: A quantitative meta analysis," *Archives of Dermatology*, vol. 139, pp. 361–367, 2003.

[8] M. E. Celebi, H. A. Kingravi, H. Iyatomi, Y. Alp Aslandogan, W. V. Stoecker, R. H. Moss, J. M. Malters, J. M. Grichnik, A. A. Marghoob, H. S. Rabinovitz, and S. W. Menzies, " Border Detection in Dermoscopy Images using Statistical Region Merging," *Skin Res. and Tech.* , vol. 14, pp. 347-353, 2008.

[9] P. G. Cavalcanti and J. Scharcanski, "Automated prescreening of pigmented skin lesions using standard cameras," *Comput Med Imag and Graph*, vol. 6, pp. 481-491, 2011.

[10] A. R. Sadri, M. Zekri, S. Sadri, N. Gheissari, M. Mokhtari and F. Kolahdouzan, "Segmentation of Dermoscopy Images using Wavelet Networks," *IEEE Trans. Biomed Engg.*, vol. 60(4), pp. 1134-1141, 2013.

[11] J. Glaister, "Automatic segmentation of skin lesions from dermatological photographs," M.S. thesis, Dept. of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada, 2013.

[12] J. Glaister, A. Wong and D. A. Clausi, "Segmentation of Skin Lesions From Digital Images Using Joint Statistical Texture Distinctiveness," *IEEE Trans. Biomed. Engg.*, vol. 61(4), pp. 1220-1230, 2014.

[13] R. Amelard, J. Glaister, A. Wong and D. A. Clausi, "High-Level Intuitive Features (HLIFs) for Intuitive Skin Lesion Description," *IEEE Trans. Biomed. Engg.,* 62(3), pp. 820-831, 2015.

[14] M. H. Jafari, S. Samavi, N. Karimi, S. M. R. Soroushmehr, K. Ward, K. Najarian, "Automatic Detection of Melanoma Using Broad Extraction of Features from Digital Images," in *Proc. 38th Annu. Int. Conf. IEEE Engg. in Medicine and Biology Society (EMBC)*, Florida, 2016, pp. 1357-1360.

[15] P. Cavalcanti, Y. Yari, J Scharcanski, "Pigmented skin lesion segmentation on macroscopic images," in *Proc. 25th Int. Conf. on Image Vision Computing*, 2010, pp 1–7.

[16] P. Cavalcanti, J. Scharcanski, C Lopes C , "Shading attenuation in human skin color images," in *Proc. 6th Int. Symp.* on *Advances in Visual Computing*, USA, 2010, pp. 190–198.

[17] Otsu N, "A threshold selection method from gray-level histograms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 9(1), pp. 62–66, 1979.

[18] M. H. Jafari, E. Nasr-Esfahani, N. Karimi, S. M. Reza Soroushmehr, S. Samavi, K. Najarian, "Extraction of skin lesions from non-dermoscopic images for surgical excision of melanoma," *Int. J. CARS*, 2017, DOI 10.1007/s11548-017-1567-8

[19] K. Korotkov and R. Garcia, "Computerized analysis of pigmented skin lesions: A review," *Artificial Intel. in Medicine*, 2012, pp. 69-90.

[20] A.Ali1, M. S. Couceiro, and A. E. Hassenian, "Melanoma Detection using Fuzzy C-Means Clustering Coupled with Mathematical Morphology," in *Proc. 14th Int. Conf.* on Hybrid Intelligent Systems, Kuwait, December, 2014, pp.73-78.

[21] Dermatology database used in MED-NODE [online]. Available:

[22] http://www.cs.rug.nl/~imaging/databases/melanoma_naevi/

[23] I. Giotis, N. Molders, S. Land, M. Biehl, M.F. Jonkman and N. Petkov, "MED-NODE: A computer-assisted melanoma diagnosis system using non-dermoscopic images," *Expert Systems with Applications*, vol. 42, pp.6578-6585, 2015.

[24] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, "Digital Image Processing using Matlab," 2nd edition, McGraw Hill Education, 2009.

[25] Q. Wu, F. Merchant, and K. R. Castleman, "Microscopic Image Processing," Academic Press, 2008.

[26] R. M. Rangayyan, "Biomedical Image Analysis," CRC Press, London, 2004.

[27] J. Somasekar, and B.E. Reddy, "Segmentation of erythrocytes infected with malaria parasites for the diagnosis using microscopy imaging," *Comput Electr Eng*, vol. 45, pp. 336-351, 2015.

[28] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *J. Cybernetics*, vol. 3, pp. 32-57, 1973.

[29] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algoritm," Plenum Press, New York, 1981.

Salam Shuleenda Devi

Salam Shuleenda Devi has completed her PhD from National Institute of Technology Silchar, India and M.Tech from KIIT University, Odisha India. She is currently working as Assistant Professor in the Department of Electronics and Communication Engineering at National Institute of Technology Meghalaya, India. Her research interests include Biomedical Image Processing, Machine Learning, and Pattern Recognition. e-mail: shuleenda26@gmail.com

Ngangbam Herojit Singh

Ngangbam Herojit Singh has completed his PhD from National Institute of Technology Manipur, India. He is currently working as Assistant Professor (temporary) in the Department of Computer Science Engineering at National Institute of Technology Mizoram, India. His major research interests are Artificial Intelligence, Robotics, and Machine Learning. e-mail: herojitng@gmail.com

Rabul Hussain Laskar

Rabul Hussain Laskar has completed his PhD from National Institute of Technology, Silchar, India and his M.Tech from Indian Institute of Technology, Guwahati. He is currently working as Associate Professor in the Department of Electronics and Communication Engineering at National Institute of Technology Silchar, India. His major research interests are in Speech processing, Image processing, Digital Signal Processing. e-mail: rhlaskar@ece.nits.ac.in

Voltage Stability Assessment of Radial Distribution Systems Including Optimal Allocation of Distributed Generators

Ali Selim[1,2], Salah Kamel[2,3], Loai S. Nasrat[2], Francisco Jurado[1]*

[1] Department of Electrical Engineering, University of Jaén, 23700 EPS Linares, Jaén (Spain)
[2] Department of Electrical Engineering, Faculty of Engineering, Aswan University, 81542 Aswan (Egypt)
[3] State Key Laboratory of Power Transmission Equipment and System Security and New Technology, Chongqing University, Chongqing 400030 (China)

Received 24 March 2019 | Accepted 14 October 2020 | Published 18 February 2020

Abstract

Assessment of power systems voltage stability is considered an important assignment for the operation and planning of power system. In this paper, a voltage stability study using Continuous Power Flow (CPF) is introduced to evaluate the impact of Distribution Generator (DG) on radial distribution systems. On the way to allocate the DG, a hybrid between the Voltage Stability Index (VSI) and Whale Optimization Algorithm (WOA) is developed. The main purpose of using VSI is to find the most sensitive buses for allocating the DG in the system. Hence, Fuzzy logic control with the Normalized VSI (NVSI) and the voltage magnitude at each bus are used to determine the candidate buses. However, the best DG size is calculated using WOA. Four standard radial distribution systems are used in this paper; 12, 33, 69, and 85-bus. The developed hybrid optimization method is compared with other existing analytical and metaheuristic optimization techniques to prove its efficiency. The results prove the ability of the developed method in the allocation of DG. In addition, the influence of the DG integration on enhancing the voltage stability through injecting the proper active and reactive powers is studied.

Keywords

Whale Optimization Algorithm, Fuzzy Logic Controller, Voltage Stability Index, Optimal DG Placement.

DOI: 10.9781/ijimai.2020.02.004

I. Introduction

RECENTLY, the complexity of the power system has been increased to align the expansion of the total load demand due to the economic and environmental constraints [1]. Power systems could be viewed as complex systems due to their structure which includes many types of components such as controlling, measuring, and monitoring devices. Hence, the increase of the load may lead to disturbance occurrence in the power system which brings an unallowable reduction in the voltage level and raises the probability of voltage instability which affects the power system operation.

Voltage stability is defined as the ability of the power system to keep the voltage of all buses at the acceptable range under abnormal conditions [1]. The probability of the voltage instability increases when there is no chance to recover the loads by the active or reactive power demand which causes the voltage collapse [2].

The voltage stability assessment has been considered as the main problem in power system monitoring [1] [3]. Considerable attention has been paid to determine maximum power system load ability limit before voltage collapse occurs [4] [5].

Voltage collapse is identified as a strongly decaying in the power system voltage level [2]. Voltage collapse could bring an unexpected shortage in the power system. Some countries challenged the voltage collapse phenomenon such as France, Japan, Sweden, and Germany [6]. Correspondingly, voltage stability indices (VSIs) have been presented to identify the most sensitive bus in the system that can lead to voltage instability [7].

To overcome the voltage instability and voltage collapse problems, many studies have been proposed and presented [8] [9]. However, the integration of distributed generators (DGs) with the power system increases power system reliability and decreases power losses. Consequently, the best solution has been addressed in the literature and is to use the active and reactive power of the DG which is connected at a suitable location to operate systems with maximum economic and reliability and enhance the performance of power system [10]. Therefore, accurate indices are required by the power systems utilities to clarify the possibility of the voltage collapse occurring, then the utilities tend to use a compensation device such as a DG [8] [11].

Currently, several research works have been presented to optimally allocate DGs in the power system. Numerous optimization techniques have been used subjected to different objective functions to increase the advantages of DGs, for instance, power loss decline, voltage profile improvement, and voltage stability [12]. DG allocation problem using optimization can be classified into many different categories based on objective functions, constraints, and the type of algorithms [13].

Two main algorithms have been frequently used in the DG allocation problem [13]. The first algorithm is the analytical method, it is a simple method that implements a mathematical formulation to

* Corresponding author.
E-mail address: fjurado@ujaen.es

maximize or minimize an objective function by changing the main control variables [14]. Many analytical indices have been applied to allocate the DG into distribution systems. The minimization of power loss using an analytical technique was presented in [15]. Loss sensitivity factor (LSF) has been used to find the most sensitive bus in the system subject to the change in the active or reactive power [16]. Furthermore, the power stability index (PSI) and voltage stability index (VSI) have been introduced to integrate DG in the distribution network. These two stability indices have been applied to recognize the most sensitive buses which could cause instability with increasing the load [17].

The second algorithm is the metaheuristic optimization algorithm, which attempts to generate all possible solutions from an initial set of solutions. A genetic algorithm (GA) has been used to optimally place the DG in distribution system [18]. Particle swarm optimization has been applied in [19]. Cuckoo Search Algorithm (CSA) has been applied in references [20] [21] to decrease the power loss and improve the voltage profile. For the multiobjective optimization problem, and NSGA II has been introduced in [22] to simultaneously minimize the power loss and maximize the VSI. Metaheuristic optimization techniques are intelligent based optimization methods [23] [24]. The main advantage of the metaheuristic techniques is the ability to handle complex problems without going so far in the problem details [13]. However, these algorithms take an uncertain time of convergence based on the search space limits and may trap in the local optima [13].

To overcome the drawback of the metaheuristic optimization technique and decrease its search space, a hybrid optimization technique can be implemented. Hence, this paper presents a hybrid between analytical VSI and a modern heuristic optimization technique named Whale Optimization Algorithm (WOA). WOA proved its efficiency as a nature-inspired based technique in comparison to the other methods [25]. The VSI is mathematically formulated, then a fuzzy logic controller is implemented using the normalized VSI and voltage magnitude at each node to arrange the system buses according to the fuzzy weighting output. The highest output weighting buses are chosen to be the candidate buses to the DG placement. Finally, the WOA is used to compute the optimal location and size of injected active and reactive powers of the DG to minimize the total power losses. The MATLAB environment is used to formulate and evaluate the objective function. The DG influence on the voltage stability is traced with PV curve analysis that performed using the continuation power flow (CPF) using PSAT package. The developed method is tested on 12, 33, 69, and 85 radial distribution systems, the results prove that using the optimal allocation of the DG increases the voltage stability.

The paper structure is prepared as follows: Section II addresses the mathematical formulation of the problem which includes the objective function, VSI, fuzzy logic controller, and WOA. The optimization process of allocating DG is introduced in Section III. The simulation results and discussion based on the study test system are presented in Section IV. Finally, Section V presents the conclusion.

II. Mathematical Problem Formulations

In this section, the main objective function and the mathematical formulation for the VSI with the fuzzy logic controller and the metaheuristic WOA are presented.

A. Objective Function

In this paper, the main purpose of the DG connection in the distribution system is to reduce the power losses P_{loss} hence, the objective function F can be expressed as.

$$F = min\ (P_{loss}) \tag{1}$$

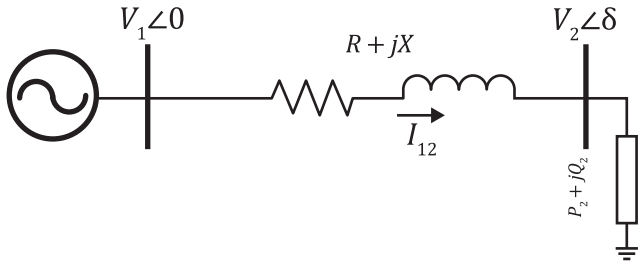


Fig. 1 Equivalent circuit model.

Where, for each two linked buses, j, the power loss P_{loss_z} can be calculated using the resistance R_{ij} and the current I_z passing through the branch z as follows:

$$P_{loss\_z} = R_{ij} I_z^2 \tag{2}$$

To calculate the overall active power loss in the distribution system, a summation of the losses in all system branches n_br are added as follows:

$$P_{loss} = \sum_{z=1}^{n\_br} P_{loss\_z} \tag{3}$$

B. Implementation of Voltage Stability Index VSI

To obtain a mathematical formulation for VSI, simple implementation for the radial distribution system consists of two buses is used as presented in Fig 1. The figure shows that the power transfers from bus 1 (where the source is connected) over a transmission line have an impedance $R_{12} + JX_{12}$ to the load connected at bus 2. The load consuming active and reactive power $P_2 + JQ_2$.

In this case, the load current at bus 2 is the same branch current passing between 1 and 2, hence the brach current I_{12} is expressed as follows:

$$I_{12} = \left[\frac{P_2 + jQ_2}{V_2\angle\delta}\right]^* \tag{4}$$

Hence, the voltage V_2 can be calculated using the voltage V_1 and the voltage drop across the line as :

$$V_2\angle\delta = V_1\angle 0 - (R_{12} + jX_{12})\ I_{12} \tag{5}$$

Using the current in (4) then eq (5) can be rewritten as:

$$V_2\angle\delta = V_1\angle 0 - (R_{12} + jX_{12})\left(\frac{P_2 - jQ_2}{V_2\angle -\delta}\right) \tag{6}$$

Reorganize the eq (6) using multiplication with $V_2\angle -\delta$

$$V_2^{\,2} = (V_1V_2\cos(\delta) - jV_1V_2\sin(\delta)) - (R_{12} + jX_{12})(P_2 - jQ_2) \tag{7}$$

Separate the real and imaginary parts

$$\begin{aligned} V_2^2 + P_2R_{12} + Q_2X_{12} &= V_1V_2\cos\delta \\ P_2X_{12} - Q_2R_{12} &= -V_1V_2\sin\delta \end{aligned} \tag{8}$$

Let $\delta \approx 0$ and $R_{12} = \frac{P_2X_{12}}{Q_2}$

$$V_2^2 - V_2V_1 + \left(\frac{P_2^2}{Q_2} + Q_2\right)X_{12} = 0 \tag{9}$$

Equation (9) can be solved as a quadratic equation and for stable bus voltages condition, $b^2 - 4ac \geq 0$.

Finally, the VSI can be obtained as follows:

$$VSI = \frac{4X}{V_1^2}\left(\frac{P_2^2}{Q_2} + Q_2\right) \leq 1 \tag{10}$$

For normal operation, the system can be considered a stable as long as the $VSI < 1$. So, the system is more stable when the VSI value is close to zero. Consequently, the node with a maximum value of the VSI is the most sensitive node in the system and should be chosen as the optimal DG location.

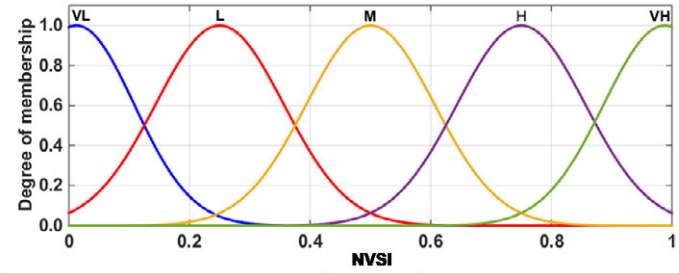
C. Candidate Buses Using the Fuzzy Logic Controller

The candidate buses are arranged using a fuzzy logic controller. The fuzzy controller is implemented with two inputs and one output. The first input is NVSI and can be expressed as follows:

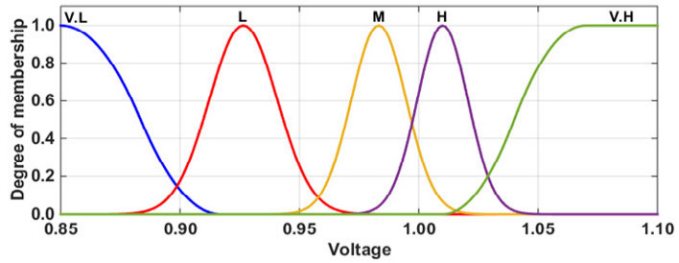
$$NVSI_i = \frac{VSI_i - VSI_{max}}{VSI_{max} - VSI_{min}} \tag{11}$$

where all $NVSI_i$ are between [0,1].

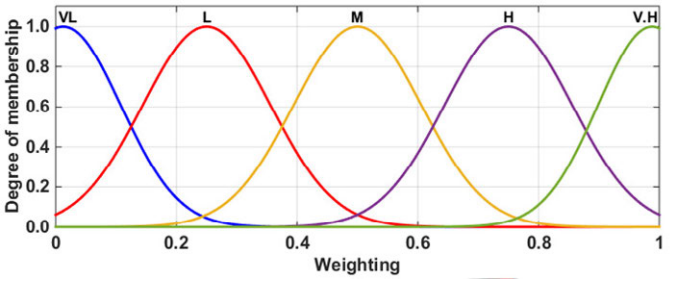
Five membership functions are utilized to represent the $NVSI_i$ as illustrated in Fig. 2.a. Also, the second input is the voltage profile at each node and represented with five membership functions as presented in Fig 2.b. The fuzzy output is calculated using predetermined rules based on (IF, Then). The output degree is shown in the surface plot in Fig. 3. Hence, each bus obtains a weighting value then these values are listed in descending order as a candidate bus.



(a) Membership function for NVSI.



(b) Membership function for voltage magnitude.



(c) Membership function for weighting.

Fig. 2. Fuzzy inputs and output memberships.
(VL= Very Low, L= Low, M= Medium, H= High, and VH= Very High)

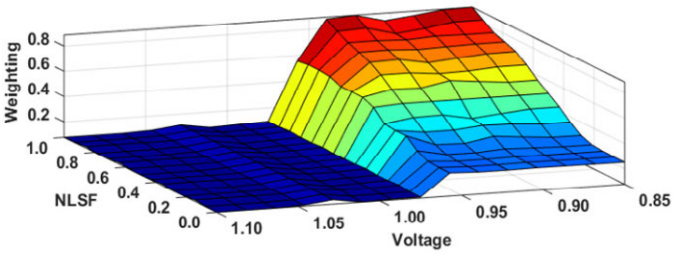


Fig. 3. Fuzzy rules output in 3 D plot.

D. Overview of Whale Optimization Algorithm (WOA)

WOA is a metaheuristic optimization technique that has been recently introduced by S. Mirjalili in [25]. The performance of WOA has been validated using 29 optimization benchmark functions and 6 structural design problems. Results show the efficiency of WOA in comparison with some nature-inspired algorithms [25]. In fact, the WOA simulates the hunting behavior of humpback whales. These whales are characterized by a bubble net feeding technique for hunting their prey.

In consequence, three stages have been utilized to mathematically represent the hunting procedure of the whales as described below:

1. Search for Prey

In order to imitate the random movement of the whales in case of their searching, a random search whale X^{rand} is used to update the position of the whales as follows:

$$X^{k+1} = X^{rand} - A.D \tag{12}$$

$$D = |C.X^{rand} - X^k| \tag{13}$$

where, X^k is the current position of the whale at iteration k, and X^{k+1} is the updated position for the next iteration. D represents the distance from the random search agent X^{rand} to X^k.

A is a random value with prespecified limits $[-a, a]$ where coefficient a is reducing from 2 to 0, then A can be expressed with the following formula:

$$A = 2\,a.r - a \tag{14}$$

$$a = 2 - k\,(2 / K_{max}) \tag{15}$$

where, r is a random number in the range of [0, 1], then C can be computed as:

$$C = 2.r \tag{16}$$

2. Encircling Prey

The mathematical formulation for encircling the prey by the whale can be adopted using the next equations:

$$X^{k+1} = X^* - A.D \tag{17}$$

$$D = |C.X^* - X^k| \tag{18}$$

The whales use a spiral-shaped in to encircle the prey, this shape can be mathematically presented as:

$$X^{k+1} = D'.e^{bl}.\cos(2\pi l) + X^* \tag{19}$$

$$D' = |X^* - X^k| \tag{20}$$

In this case, D' describes the absolute distance value between X^k (current whale position) and the best position X^*. b implements the formula of the logarithmic spiral. l is a random number with a range [-1, 1].

3. Bubble Net Hunting

Two whale movements have been used to formulate the bubble net hunting, the first movement uses the encircling prey by applying (17), where the second uses the spiral shape as described in (19). The transition between those two movements can occur using a random probability parameter p as follows:

$$X^{k+1} = \begin{cases} X^* - A.D, & p < 0.5 \\ D'.e^{bl}.\cos(2\pi l) + X^*, & p \geq 0.5 \end{cases} \quad (21)$$

The overall WOA is presented in the pseudocode shown in Fig. 4.

Initialize a set of random search whales $X_i = (X_1, X_2, ..., X_n)$.
within the limits $X_L \leq X_i \leq X^U$
Calculate the objective function for each search whale
Store the best solution X^*
While $(k < K_{max})$
for each search whale X_i
Update the parameters $A, a, and\ C$ using (14), (15), and (16) respectively
$p = rand$
$l = rand$
If 1 $p < 0.5$
If 2 $|A| < 1$
Update the whale's positions using (17), and (18)
else if 2 $|A| \geq 1$
Choose a random whale X^{rand}
Update the whale's positions using (12), and (13)
end if 2
else If 1 $p \geq 0.5$
Update the whale's positions using (19), and (20)
end if 1
Calculate the objective function
Update the best solution X^*
$K = K + 1$
end while
return the final best solution stored X^*

Fig. 4. Pseudocode of WOA.

III. Voltage Stability Assessment with DG Allocation

The assessment of the power system voltage stability with optimal allocation of the DG is described in the flowchart shown in Fig. 5. The overall process is concluded in the main following steps:

Step 1: Read the system data (line data and load data) and define the objective function.

Step 2: Run the power flow and obtain the voltage magnitude and angle for all buses.

Step 3: Compute the VSI for each node and calculate the $NVSI_i$.

Step 4: Apply the fuzzy logic controller to arrange the candidate buses.

Step 5: Randomly initialize a set of search whales with the candidate buses, WOA parameters, and Max. number of iterations K_{max}.

Step 6: Run power flow and calculate the objective function for each search whale and store the best solution.

Step 7: For each search whale, update the parameters a, A, C, l *and* p.

Step 8: Calculate the objective function for each search agent.

Step 9: Update the best solution.

Step 10: If $k < K_{max}$, repeat Step 5.

Step 11: Return the stored best solution obtained so far.

Step 12: Use the best solution as the optimal size and location for the DG.

Step 13: Run the continuation power flow and assess the power system voltage stability by tracing the PV curve.

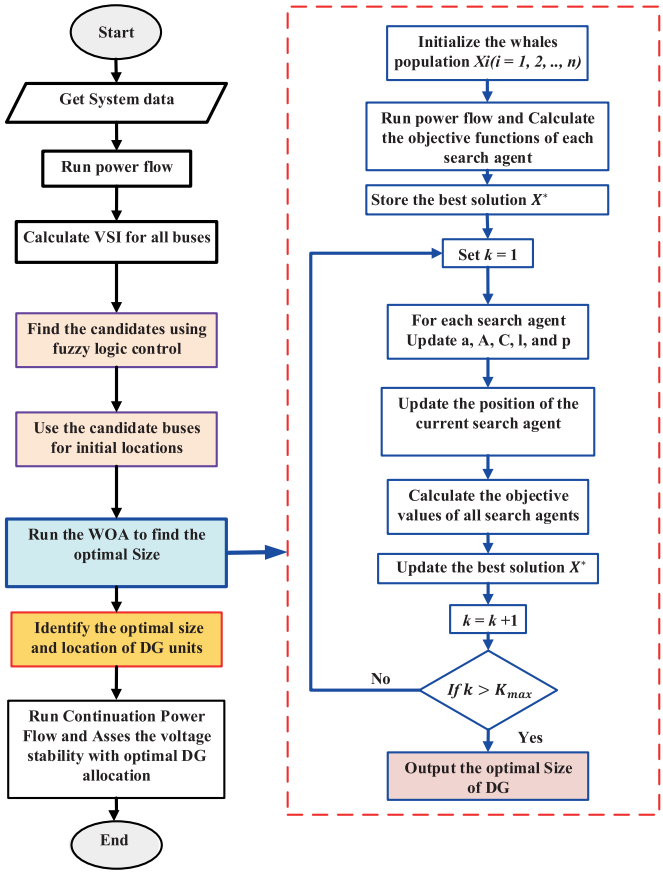


Fig. 5. Flowchart for voltage stability assessment with optimal DG allocation.

IV. Results and Discussion

In this section, the performance of the proposed hybrid VSI and WOA technique is tested using 12, 33, 69, and 85 radial distribution systems. The feasibility and efficiency of hybrid VSI and WOA to optimally allocate DG in radial distribution systems are proved compared with other well-known optimization techniques. All simulations have been carried out using MATLAB M-files. WOA parameters are set as; the number of agents = 100, the maximum number of iterations = 50. The following four cases have been considered for the DG allocation and voltage stability assessment in the studied systems:

1. Case 1: No DG unit is connected.
2. Case 2: One DG unit that injects only active power is connected.
3. Case 3: One DG unit that injects only reactive power s connected.
4. Case 4: One DG unit that injects active and reactive powers is connected.

A. 12-bus Radial Distribution System

The distribution system can be found in [26]. The system consists of 12 buses and 11 branches. The fuzzy logic controller using the NVSI

and voltage profile is applied to the 12-bus system. Table I summarizes the ordering of the candidate buses. For each branch, VSI is calculated then the NVSI is adopted as the first input of the fuzzy logic controller. The voltage at the end of each branch bus is used as the second input as presented in the table. As seen in the table, branch 8 has the highest VSI and NVSI and it receives bus 9 that has 0.9473 p.u voltage. Respecting to the fuzzy rules this bus is considered the highest output hence, the fuzzy weighting output is 0.8210. The same process is performed in all buses and the order of candidate buses is shown in Table I.

TABLE I. Candidate Bus Ordering Using Fuzzy Logic Control

Branch ID	VSI	NVSI	Receive Bus	Voltage	Fuzzy output	Bus order
1	0.00180	0.334	2	0.9943	0.0888	9
2	0.00138	0.233	3	0.9890	0.0806	11
3	0.00325	0.674	4	0.9806	0.0859	10
4	0.00274	0.555	5	0.9698	0.1314	7
5	0.00067	0.065	6	0.9665	0.1264	8
6	0.00162	0.291	7	0.9637	0.2178	6
7	0.00389	0.826	8	0.9553	0.6337	2
8	0.00463	1.000	9	0.9473	0.8210	1
9	0.00213	0.412	10	0.9445	0.4003	3
10	0.00132	0.220	11	0.9436	0.2710	4
11	0.00039	0.000	12	0.9434	0.2534	5

1. DG Allocation

In the base case (Case 1), the power flow results of the 12-bus system indicate that the active and reactive power losses are 20.7138 kW and 8.0411 kVAR, respectively. The minimum voltage bus is reported at bus 12 with 0.9434 p.u. After applying the fuzzy logic controller using NVSI and the voltage magnitude, the WOA is applied to find the final size at the candidates' buses which give the minimum power loss.

Table II gives the optimal sizes and locations of three DG cases. For Case 2, the optimal size is calculated using the WOA and its value is 235.5 kW, as shown in Table II, and the total power loss decreases to 10.774 kW with Loss Reduction (LR) reaching 47.98 %. The minimum voltage, in this case, is 0.9835 at bus 7.

At Case 3, where 210.21 kVAR reactive power is injected at bus 9, the LR is 39.25% and the minimum voltage reports at bus 12 with 0.9563 p. u. A significant LR is obtained in Case 4 due to the active and reactive power injection. The LR is 84.76 % and the minimum voltage bus is 7 with 0.9907 p.u. Overall enhancement in the voltage profile in the three cases is illustrated in Fig 6. A considerable enhancement is achieved at Case 4.

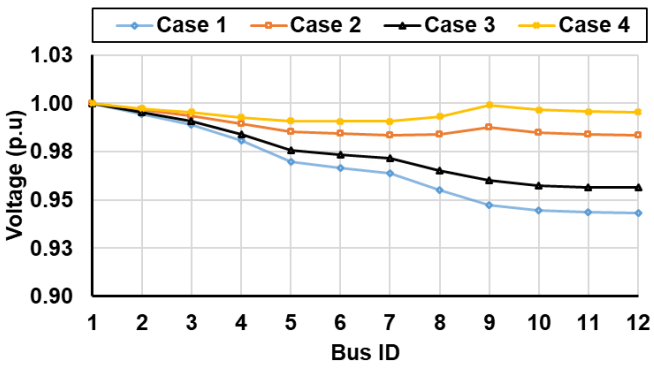


Fig. 6. Voltage profile of 12-bus at different case studies.

The convergence characteristic for the hybrid VSI and WOA for the three case studies (Case 2, Case 3, and Case 4) is presented in Fig. 7. It is clear that WOA converged fast due to decreasing the search space through the fuzzy process applied with the VSI.

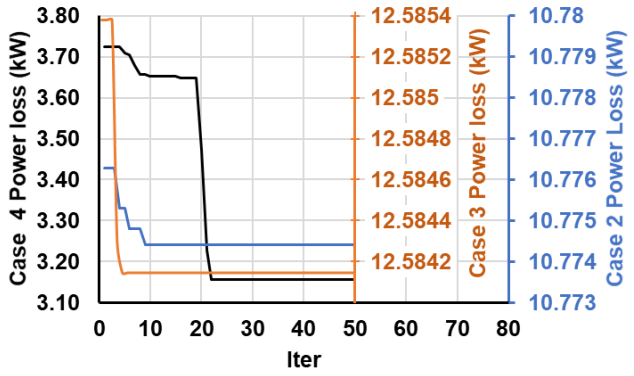


Fig. 7. Convergence characteristics of the WOA for 12-bus at different case studies.

TABLE II. DG Allocation in 12- bus System at Different Case Studies

		Case 1	Case 2	Case 3	Case 4
DG Location		-	9	9	9
DG Size	kW	-	235.50	-	230.82
	kVAR	-	-	210.21	213.93
PL (kW)		20.714	10.774	12.584	3.157
QL (kVAR)		8.041	4.125	4.824	1.108
Min bus voltage		12	7	12	7
Vmin (p.u)		0.9434	0.9835	0.9563	0.9907
LR %		0.00	47.98	39.25	84.76
Lambda max		5.31	5.96	6.54	6.31

2. Voltage Stability Assessment

In this section, the PV curve is obtained with the CPF using the PSAT package to evaluate the voltage stability of the radial distribution system with DG allocation. Fig. 8 shows the PV curves for the four case studies at bus 12 which is the minimum bus voltage. It can observe that in Case 1 the maximum lambda loading is 5.31%. In Case 2, the loading factor increases to 5.96%, which demonstrates the influence of the active power of the DG. However, the maximum loading lambda is achieved at Case 3 which equals 6.54% and this is due to the injected DG reactive power. Finally, the impact of the active and reactive power injected by the DG on the PV curve is carried out in Case 4 and the reported Lambda max is 6.31%.

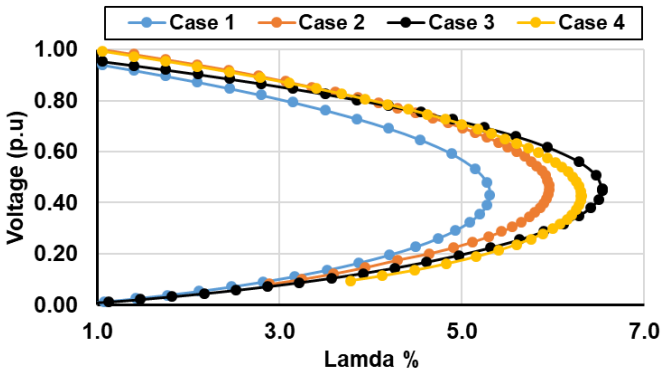


Fig. 8. PV curve for 12-bus at different case studies.

B. 33-Bus Radial Distribution System

The second system used to validate the proposed method is the 33-bus radial distribution system. The complete depiction of this test system involves the line and load data that can be obtained in [27]. The fuzzy logic controller with the NVSI and voltage magnitude is applied to this system and the highest candidates' buses are 7 and 30. However, during the search process with the WOA and the updating positions, buses 6 and 30 are found to be the best buses for DG allocation as summarized in Table III.

1. DG Allocation

Table III presents the optimal sizes and locations of the three cases of DG allocations. In Case 1, the power loss is 210.986kW and it decreases to 111.019 kW, 151.365 kW, and 67.855 kW at Case 2, Case 3, and Case 4, respectively. The voltage profile has a significant enhancement in Case 4 as shown in Fig. 9 where the minimum voltage is reported at bus 18, which equals 0.9584 p.u.

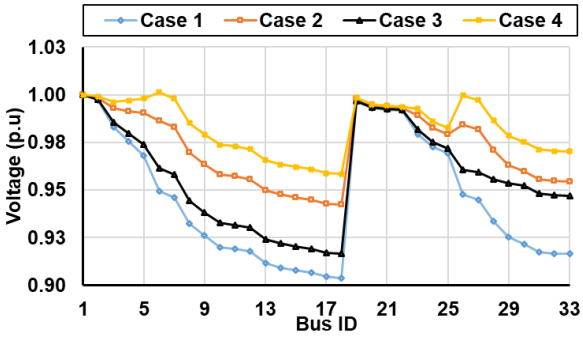


Fig. 9. Voltage profile of 33-bus at different case studies.

The convergence characteristics of the proposed method are displayed in Fig 10. Case 4 needed the highest number of iterations to converge because it searches for the active and reactive powers to minimize the power loss.

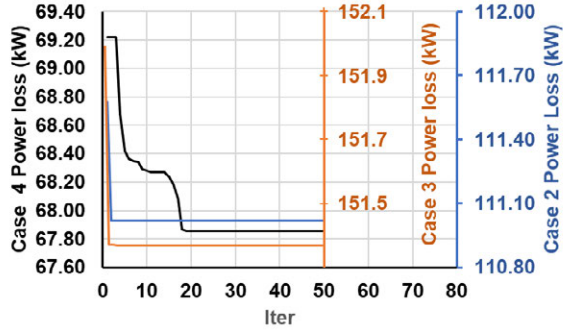


Fig. 10. Convergence characteristics of the WOA for 33-bus at different case studies.

TABLE III. DG Allocation in 33-Bus System at Different Case Studies

		Case 1	Case 2	Case 3	Case 4
DG Location		-	6	30	6
DG Size	kW	-	2590.21	-	2558.92
	kVAR	-	-	1258.013	1760.70
PL (kW)		210.986	111.019	151.365	67.855
QL (kVAR)		143.127	81.717	103.890	54.840
Min bus voltage		18	18	18	18
Vmin (p.u)		0.9038	0.9424	0.9165	0.9584
LR %		0.00	47.38	28.26	67.84
Lambda max		3.40	3.72	3.90	3.86

2. Voltage Stability Assessment

The PV curve for bus 18, which is the minimum bus voltage at Case 1, is exhibited in Fig. 11 in different case studies. From this figure, it can be observed that the highest lambda is achieved at Case 3 which reaches 3.90% where it was 3.40% in Case 1. The lambda max is 3.72 % and 3.86 % at Case 2 and Case 4, respectively.

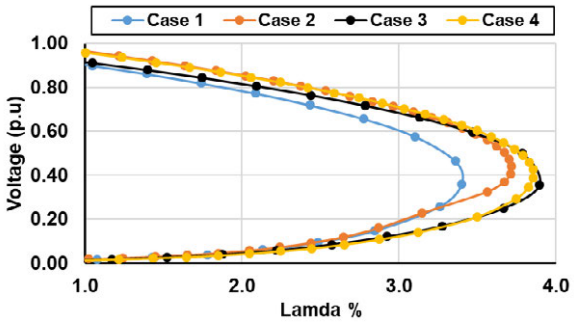


Fig. 11. PV curve for 33-bus at different case studies.

C. 69-Bus Radial Distribution System

This system consists of 69 radial connected buses and 68 branches, the single line diagram and the line and load data is available in [28]. In this system, the highest order candidate bus obtained by the fuzzy logic controller is bus 61, hence it is chosen by the WOA to be the optimal location for the DG in the three case studies.

1. DG Allocation

Table IV presents the base case results of the 69-bus system where the power loss is 224.95 kW, the reactive power loss is 102.146 kVAR, and the minimum voltage is 0.9092 at bus 65. However, with connected three DG types at three case studies, the active power LR reaches 63.02, 32.43, and 89.71 at Case 2, Case 3, and Case 4, respectively. Also, the voltage profile achieves a significant improvement as shown in Fig. 12. Fig 13 demonstrates the convergence characteristics of the proposed method and it is clear that the WOA converged at a low number of iterations.

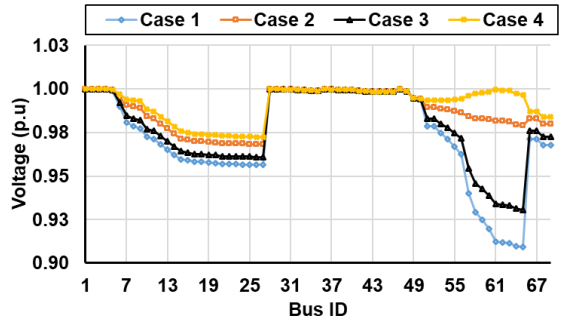


Fig. 12. Voltage profile of 69-bus at different case studies.

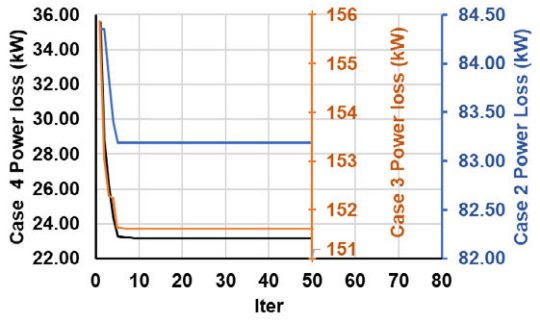


Fig. 13. Convergence characteristics of the WOA for 69-bus at different case studies.

TABLE IV. DG ALLOCATION IN 69- BUS SYSTEM AT DIFFERENT CASE STUDIES

		Case 1	Case 2	Case 3	Case 4
DG Location		-	61	61	61
DG Size	kW	-	1872.63	-	1828.24
	kVAR	-	-	1329.90	1300.73
PL (kW)		224.950	83.189	152.005	23.146
QL (kVAR)		102.146	40.522	70.489	14.370
Min bus voltage		65	27	65	27
Vmin (p.u)		0.9092	0.9683	0.9307	0.9725
LR %		0.00	63.02	32.43	89.71
Lambda max		3.21	3.95	4.02	4.24

2. Voltage Stability Assessment

Fig. 14 reveals the PV curves for bus 65 at different case studies, the highest lambda max is 4.24 % attained at Case 4 where the impact of the DG integration on the voltage stability is clearly assessed using the PV curve.

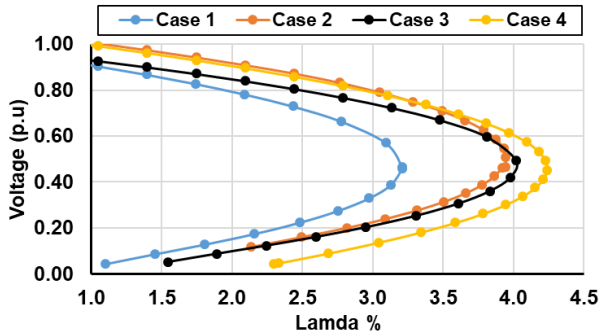


Fig. 14. PV curve for 69-bus at different case studies.

D. 85-Bus Radial Distribution System

This system considers a large and long radial distribution system. The system includes 85 buses and 84 branches and the overall data can be obtained from [29]. Bus 8 has the highest order weighting based on the fuzzy logic controller so that this bus is considered the optimal location for DG location at the different case studies.

1. DG Allocation

Due to the length of this system, the power loss is 315.973 kW and the lowest voltage is 0.8714 p.u at bus 54 as given in Table V. By applying the proposed VSI and WOA at three DG allocation cases, the Power loss decrease to 175.470, kW 180.548 kW, and 62.576 kW with LR 44.47%, 42.86 %, and 80.20 % at Case 2, Case 3, and Case 4 respectively. Also, the voltage at bus 54 increases to 0.9283 p.u, 0.9108 p.u, and 0.9616 p.u at Case 2, Case 3, and Case 4, respectively. The enhancement in the voltage profiles for all buses is presented in Fig. 15.

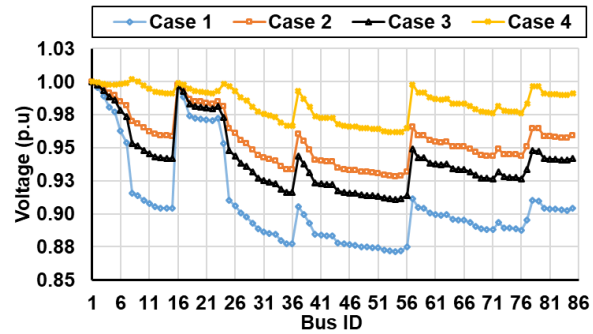


Fig. 15. Voltage profile of 85-bus at different case studies.

Fig. 16 proves the efficiency of the WOA in allocating the DG at different case studies in the large-scale distribution system with low iteration numbers.

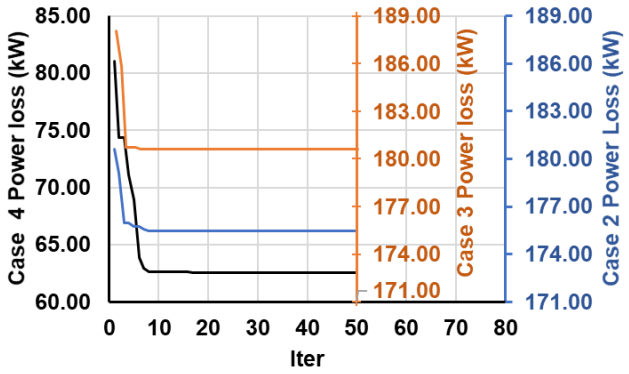


Fig. 16. Convergence characteristics of the WOA for 85-bus at different case studies.

TABLE V. DG ALLOCATION IN 85- BUS SYSTEM AT DIFFERENT CASE STUDIES

		Case 1	Case 2	Case 3	Case 4
DG Location		-	8	8	8
DG Size	kW	-	2373.78	-	2268.50
	kVAR	-	-	2332.77	2259.11
PL (kW)		315.973	175.470	180.548	62.576
QL (kVAR)		198.701	104.385	107.276	27.997
Min bus voltage		54	54	54	54
Vmin (p.u)		0.8714	0.9283	0.9108	0.9616
LR %		0.00	44.47	42.86	80.20
Lambda max		2.55	2.74	3.29	2.94

2. Voltage Stability Assessment

Similarly, the PV curve for the 85-bus is drawn using the CPF at different case studies of DG integration as exhibited in Fig. 17. The base case lambda max of this system is 2.55 % and this value increased to 2.74 % when connecting DG at bus 8 with 2373.78 kW. Furthermore, a considerable increase in the lambda is accomplished at Case 2 with 2332.77 kVAR connected on bus 8. Finally, at Case 4 the lambda reaches 2.94 % with 2268.5 kW and 2259.11 kVAR at the same bus.

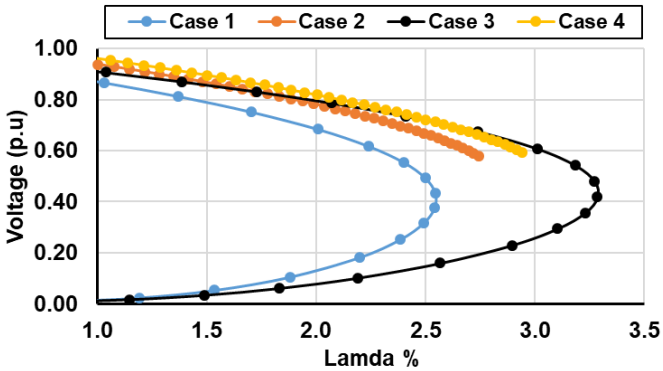


Fig. 17. PV curve for 85-bus at different case studies.

E. Result comparison

To demonstrate the efficiency of the developed method, a comparison is performed with other optimization techniques mentioned in the literature such as; analytical technique [15], PSO [12], CS [21],

NSGA-II [22]. As presented in Table VI, the proposed method has the highest LR compared to all techniques. However, for the maximum loading factor the proposed method gives the highest loading factor in 12- bus that equals 6.31 %. Also, the same lambda max in 69-bus is given by the proposed method and NSGA II which reaches 4.24%. The maximum loading factor in 33-bus is given by the CS algorithm. In addition, for 85-bus the NSGA II has the maximum lambda 2.94 %.

TABLE VI. Result Comparison with Other Optimization Techniques

Method		12-bus	33-bus	69-bus	85-bus
Proposed method VSI and WOA	DG location	9	6	61	8
	DG size (kVA)	314.71	3106.14	2243.73	3201.5
	LR %	**84.76**	**67.84**	**89.71**	**80.20**
	Lambda max %	**6.31**	3.86	4.24	2.94
Analytical technique [15]	DG location	9	6	61	8
	DG size (kVA)	227.15	2490.78	1807.8	22088.6
	LR %	47.95	47.31	62.95	44.28
	Lambda max %	5.92	3.70	3.92	2.88
NSGA-II [22]	DG location	8	7	61	25
	DG size (kVA)	435	3715	2663.8	2484.5
	LR %	33.97	35.92	53.8	42.75
	Lambda max %	6.08	3.88	**4.24**	**2.96**
PSO [12]	DG location	9	7	61	
	DG size (kVA)	253.9	2895.1	2026.4	
	LR %	47.7	45.55	62.65	
	Lambda max %	6.03	3.78	4.03	
CS [21]	DG location		61		
	DG size (kVA)		2200		
	LR %		62.8		
	Lambda max %		**4.06**		

V. Conclusion

In this paper, an assessment of power systems voltage stability with DG integration using CPF has been presented. A hybrid between VSI and WOA has been developed to place the DG in the radial distribution system. NVSI and voltage magnitude at each node has been modeled with the fuzzy logic controller to find the candidate buses which are the most sensitive buses to allocate the DG. Finally, the optimal size and location have been achieved using the WOA. The developed method has been tested using the 12, 33, 69 and 85 bus radial distribution systems. To present the effectiveness of the developed method, a comparison with existing techniques has been accomplished. The obtained results proved the efficiency and capability of the developed optimization technique for selecting the optimal DG location and size using the voltage stability index. In addition, a significant increase in the loadability has been obtained with integrating DG that injects both active and reactive powers. The extended work of this paper is to improve the exploration and exploitation phases of the WOA by hybridizing with other metaheuristic optimization techniques.

Acknowledgment

The authors gratefully acknowledge the contribution of the NSFC (China)-ASRT (Egypt) Joint Research Fund, Project No. 51861145406 for providing partial research funding to the work reported in this research.

References

[1] J. Modarresi, E. Gholipour, and A. Khodabakhshian, "A comprehensive review of the voltage stability indices," Renewable and Sustainable Energy Reviews, vol. 63, pp. 1-12, 2016.

[2] A. Sode-Yome and N. Mithulananthan, "Comparison of shunt capacitor, SVC and STATCOM in static voltage stability margin enhancement," International Journal of Electrical Engineering Education, vol. 41, no. 2, pp. 158-171, 2004.

[3] I. Musirin and T. A. Rahman, "Novel fast voltage stability index (FVSI) for voltage stability analysis in power transmission system," in Research and Development, 2002. SCOReD 2002. Student Conference on, 2002: IEEE, pp. 265-268.

[4] W. Ahmed, A. Selim, S. Kamel, J. Yu, and F. Jurado, "Probabilistic Load Flow Solution Considering Optimal Allocation of SVC in Radial Distribution System," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 3, pp. 152-161, 2018.

[5] M. Abdel-Nasser, K. Mahmoud, and H. Kashef, "A Novel Smart Grid State Estimation Method Based on Neural Networks," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 1, pp. 92-100, 2018.

[6] C. Reis and F. M. Barbosa, "A comparison of voltage stability indices," in Electrotechnical Conference, 2006. MELECON 2006. IEEE Mediterranean, 2006: IEEE, pp. 1007-1010.

[7] V. Murty and A. Kumar, "Optimal placement of DG in radial distribution systems based on new voltage stability index under load growth," International Journal of Electrical Power & Energy Systems, vol. 69, pp. 246-256, 2015.

[8] T. J. Overbye and C. L. DeMarco, "Improved techniques for power system voltage stability assessment using energy methods," IEEE Transactions on Power Systems, vol. 6, no. 4, pp. 1446-1452, 1991.

[9] E. Vaahedi, J. Tamby, Y. Mansour, W. Li, and D. Sun, "Large scale voltage stability constrained optimal VAr planning and voltage stability applications using existing OPF/optimal VAr planning tools," IEEE Transactions on Power Systems, vol. 14, no. 1, pp. 65-74, 1999.

[10] R. Al Abri, E. F. El-Saadany, and Y. M. Atwa, "Optimal placement and sizing method to improve the voltage stability margin in a distribution system using distributed generation," IEEE transactions on power systems, vol. 28, no. 1, pp. 326-334, 2013.

[11] M. Aman, G. Jasmon, A. Bakar, and H. Mokhlis, "A new approach for optimum simultaneous multi-DG distributed generation Units placement and sizing based on maximization of system loadability using HPSO (hybrid particle swarm optimization) algorithm," Energy, vol. 66, pp. 202-215, 2014.

[12] M. M. Aman, G. B. Jasmon, A. H. A. Bakar, and H. Mokhlis, "A new approach for optimum DG placement and sizing based on voltage stability maximization and minimization of power losses," Energy Conversion and Management, vol. 70, pp. 202- 210, 2013.

[13] M. P. HA, P. D. Huy, and V. K. Ramachandaramurthy, "A review of the optimal allocation of distributed generation: Objectives, constraints, methods, and algorithms," Renewable and Sustainable Energy Reviews, vol. 75, pp. 293-312, 2017.

[14] Ehsan, Ali, and Q. Yang. "Optimal integration and planning of renewable distributed generation in the power distribution networks: A review of analytical techniques," Applied Energy 210 (2018): 44-59.

[15] T. Gözel and M. H. Hocaoglu, "An analytical method for the sizing and siting of distributed generators in radial systems," Electric Power Systems Research, vol. 79, pp. 912-918, 2009.

[16] D. Q. Hung and N. Mithulananthan, "Multiple distributed generator placement in primary distribution networks for loss reduction," IEEE Transactions on industrial electronics, vol. 60, no. 4, pp. 1700-1708, 2013.

[17] V. Murty and A. Kumar, "Optimal placement of DG in radial distribution systems based on new voltage stability index under load growth," International Journal of Electrical Power & Energy Systems, vol. 69, pp. 246-256, 2015.

[18] R. K. Singh and S. K. Goswami. "Optimum siting and sizing of distributed generations in radial and networked systems," Electric Power Components and Systems, vol. 37, no. 2, pp. 127-145, 2009.

[19] K. Satish, V. Kumar, and B. Tyagi. "Optimal placement of different type of DG sources in distribution networks," International Journal of Electrical Power & Energy Systems, vol. 53, pp. 752-760, 2013.

[20] Moravej, Zahra, and A. Akhlaghi. "A novel approach based on cuckoo search for DG allocation in distribution network," International Journal of Electrical Power & Energy Systems, vol. 44, no. 1, pp. 672-679, 2013.

[21] Tan, W. S., et al. "Allocation and sizing of DG using cuckoo search algorithm," 2012 IEEE international conference on power and energy (PEcon). IEEE, 2012.

[22] Mohammedi, R. D., et al. "Optimal DG placement and sizing in radial distribution systems using NSGA-II for power loss minimization and voltage stability enhancement," International Review of Electrical Engineering, vol. 8, no. 06, 2013.

[23] A. Selim, S. Kamel, and F. Jurado, "Voltage Profile Improvement in Active Distribution Networks Using Hybrid WOA-SCA Optimization Algorithm," in 2018 Twentieth International Middle East Power Systems Conference (MEPCON), 2018: IEEE, pp. 1064-1068.

[24] A. Selim, S. Kamel, and F. Jurado, "Hybrid Optimization Technique for Optimal Placement of DG and D-STATCOM in Distribution Networks," in 2018 Twentieth International Middle East Power Systems Conference (MEPCON), 2018: IEEE, pp. 689-693.

[25] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in Engineering Software, vol. 95, pp. 51-67, 2016.

[26] P. R. Babu, C. Rakesh, M. N. Kumar, G. Srikanth, and D. P. Reddy, "A novel approach for solving distribution networks," in India Conference (INDICON), 2009 Annual IEEE, 2009: IEEE, pp. 1-5.

[27] R. Rajaram, K. S. Kumar, and N. Rajasekar, "Power system reconfiguration in a radial distribution network for reducing losses and to improve voltage profile using modified plant growth simulation algorithm with Distributed Generation (DG)," Energy Reports, vol. 1, pp. 116-122, 2015.

[28] M. E. Baran and F. F. Wu, "Optimal sizing of capacitors placed on a radial distribution system," Power Delivery, IEEE Transactions on, vol. 4, pp. 735-743, 1989.

[29] D. Das, D. P. Kothari, and A. Kalam, "Simple and efficient method for load flow solution of radial distribution networks," International Journal of Electrical Power & Energy Systems, vol. 17, pp. 335-346, 1995.

Ali Selim

Ali Selim received his BSc and MSc degrees in Electrical Engineering from Aswan University, Egypt, in 2010 and 2016 respectively. He is currently a Ph.D. student in the Department of Electrical Engineering at the University of Jaén (Universidad de Jaén), Spain. His research interests include mathematical optimization, planning, and control of power systems, renewable energies, energy storage, and smart grids.

Salah Kamel

Salah Kamel received the international Ph.D. degree from University of Jaen, Spain (Main) and Aalborg University, Denmark (Host) in Jan. 2014. He is an Assistant Professor in the Electrical Engineering Department, Aswan University. Also, He is a Leader for a power systems research group in the Advanced Power Systems Research Laboratory (APSR Lab), Aswan, Egypt. He is currently a Postdoctoral Research Fellow in State Key Laboratory of Power Transmission Equipment and System Security and New Technology, School of Electrical Engineering, Chongqing University, Chongqing, China. His research activities include power system modeling, analysis and simulation, and applications of power electronics to power systems and power quality.

Loai. S. Nasrat

Loai. S. Nasrat is a professor at Aswan University, Faculty of Eng., Elect. P&M Department. He has been actively involved in both basic and applied research in the area of HV engineering. His research interests include HV insulators, Nano polymeric materials, HV cables insulations, Aged oil transformer, and environmental studies. Author and co-author of more than 60 papers on HV polymeric insulating materials, Nano insulating materials published in technical journals and proceedings of national and international conferences. Dr. Loai is a member of the Egyptian Sub-Committee of CIGRE

Francisco Jurado

Francisco Jurado obtained the MSc and Ph.D. degrees from the UNED, Madrid, Spain, in 1995 and 1999 respectively. He is Full Professor at the Department of Electrical Engineering of the University of Jaén, Spain. His research activities have focused on two topics: power systems and renewable energy.

Mamdani Fuzzy Expert System Based Directional Relaying Approach for Six-Phase Transmission Line

A. Naresh Kumar[1]*, Ch. Sanjay[2], M. Chakravarthy[3]

[1] Department of Electrical and Electronics Engineering, Institute of Aeronautical Engineering, Hyderabad (India)
[2] Department of Mechanical Engineering, GITAM University, Hyderabad (India)
[3] Department of Electrical and Electronics Engineering, Vasavi College Engineering, Hyderabad (India)

Received 14 December 2018 | Accepted 29 May 2019 | Published 21 June 2019

Abstract

Traditional directional relaying methods for 6-phase transmission lines have complex effort, and so there is still a need for novel direction relaying estimation scheme. This study presents a Mamdani-fuzzy expert system (MFES) approach for discriminating faulty section/zone, classifying faults and locating faults in 6-phase transmission lines. The 6-phase fundamental component of currents, voltages and phase angles are captured at single bus and are used in the protection scheme. Simulation results substantiate that the protection scheme is very successful against many parameters such as different fault types, fault resistances, transmission line fault locations and inception angles. A large number of fault case studies have been carried out to evaluate reach setting and % error of proposed method. It provides primary protection to transmission line length and also offers backup protection for a reverse section of transmission line. The experimental results show that the scheme performs better than the other schemes.

Keywords

Fault Protection, Fuzzy Systems, Simulation, Transmission Lines.

DOI: 10.9781/ijimai.2019.06.002

I. Introduction

THE demand for electrical power is always increasing, as a rule of thumb, it doubles every decade. In order to reach this booming demand, power transfer capability of the transmission lines is required to be improved continuously. It can be done either by enhancing the voltage handling capability or by upgrading the existing power transmission systems to multi-phase (i.e. 6, 9 and 12-phase) transmission systems. The multi-phase transmission serves as an optimal solution to handle high power transfer capacity of existing transmission systems. Existing single and double circuit 3-phase transmission lines can easily be modified to operate as a single circuit 6-phase transmission line. Apart from improved power transfer capacity, 6-phase transmission systems provide several other benefits over traditional 3-phase double circuit transmission line viz. higher efficiency, lesser corona, higher thermal loading capability, better surge impedance, good voltage regulation, lesser audible noise levels, lesser radio interference and lesser conductor surface gradient. The 6-phase line theory and benefits are described in [1]-[2]. Existing 115kV, double circuit 3-phase transmission line was modified into 93kV, 2.4 km,6-phase line configuration in between Goudey-Okadale substations. The 93kV, configuration research was conducted for three years from 1992 to 1995, after it was replaced to its original 115kV, double circuit 3-phase transmission line configuration [3] because of the failure of 6-phase line protection. Although 6-phase line offers more benefits to electrical power system operations, the fault protection task in 6-phase line is more difficult than in double circuit 3-phase transmission lines. It can cause permanent damage to the transmission lines. Thus, early identification is needed to minimize the incidences severity of power system damages.

Till now, several techniques have been focusing on locating and classifying faults in 6-phase transmission line [4]-[8]. In [4], the fundamental component of measured currents is used for fault detection and classification. In [5], the Haar Wavelet Transform measured currents are employed for fault detection and classification. A methodology based on current measurements for locating faults is provided in [6]. Modular artificial neural network (ANN) scheme is demonstrated [7] for fault classification and location based on fundamental component voltages and currents. Same Modular ANN scheme is addressed [8] for fault classification and location based on wavelet transform voltages and currents. Nonetheless, all the aforementioned schemes are relevant to normal fault classification and location but they are not applicable for directional relaying. Only a few researchers have illustrated directional relaying in 6-phase transmission lines [9]–[10], but these methods do not classify and locate the faults. In [11], a neuro-wavelet algorithm for zone/section identification and fault location in 6-phase transmission lines is suggested. The directional relaying algorithms are developed using linear discriminant analysis [12], adaptive neuro-fuzzy inference system [13], decision tree [14] and ANN [15] methods. Among all the soft computing methods, fuzzy expert system (FES) is employed mostly in transmission line protection applications owing to less computation work and easy implementation, unlike other soft computing techniques. Some notable works have been reported in the literature on the soft computing techniques [16]–[18]. Fuzzy with single end directional relaying scheme for transmission line compensated by fixed series capacitor have been

* Corresponding author.
E-mail address: ankamnaresh29@gmail.com

addressed in [19]. Some studies using FES for fault classification and location in transmission lines have been published [20]-[23]. Fuzzy based model with better performance of directional relaying on double circuit transmission lines has been introduced in [24]. However, none of the authors have given the solution to discriminate faults in 6-phase transmission lines without complex effort considering the fundamental currents and MFES scheme. Therefore, a novel zone identification scheme using fuzzy inference system is presented in this paper. This paper will further develop to deal with fault location and classification.

The structure of the paper is as follows. Section II describes the transmission line system. In Section III, a Mamdani fuzzy idea for fault direction, classification and location is described. The simulation results and discussions are explained in Section IV. Finally, in Section V, conclusive remarks are given.

II. Transmission Line

As depicted in Fig. 1, a single line diagram of 6-phase transmission line connecting between two substations namely Springdale and McCalmont is used to clarify the presented schemes. The network consists of a 138 kV, 60 Hz transmission line of length 200 km divided into two zones of 100 km each. The relay is located at bus-2 which gives protection to the total transmission line. The distributed line parameter blocks are employed to model 6-phase line system while fault breakers are used to simulate faults. The zone between the sending terminal and the relaying point is considered as reverse section (zone-1) and the zone between the relaying point and the receiving terminal is the forward section (zone-2). The 6-phase line is simulated using Simulink® and Simpowersystem® toolboxes of MATLAB.

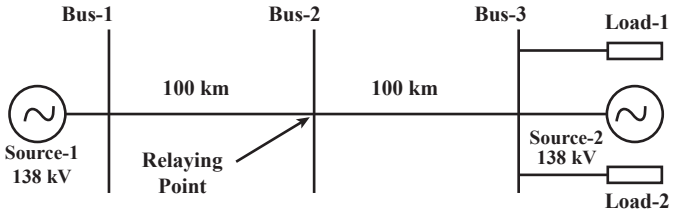


Fig. 1. Diagram of the studied system.

Fig. 2 represents the six phase voltages of D-g fault at 46 ms of time at 42 km for fault resistance of 10 Ω. Fig. 3 sketches the six phase currents of D-g fault at 46 ms of time at 42 km for fault resistance of 10 Ω. The zero sequence current during D-g fault is shown in Fig. 4. The phase angle of positive sequence current during D-g fault in the forward zone is seen in Fig. 5. The recorded instantaneous 6-phase currents and voltages are applied to a 2nd order low pass Butterworth filter. The filter cut-off frequency is kept at 480 Hz. After that, these currents and voltages are sampled with a frequency of 1.2 kHz. This frequency has been chosen based on the Nyquist sampling criteria. Then, these signals are fed to one full cycle discrete fourier transform for estimation of the fundamental components of 6-phase currents and voltages. Now, these currents and voltages are normalized in between -1 to +1. From the fundamental component of currents and voltages, the fundamental component of impedances is obtained using equation (1). From the fundamental component of impedances, 35 post-fault cases are selected after the full cycle of fault occurrence. Mean impedance is computed using equation (2), which changes from position to position for various types of fault.

$$Z = Vf/If \quad (1)$$

Where Z - fundamental component of impedance,
Vf - fundamental component of voltage and
If - fundamental component of current

$$Zm = \sum_{k=1}^{n} Zk/n \quad (2)$$

Where Zm - Mean fundamental component of impedance and
n - Number of fault samples chosen.

To analyze the currents and voltages of the line during faulty condition, consider a single phase to ground fault (D-g) has occurred at 46 ms time. Before the inception of fault, the 6-phase currents and voltages in all the phases are same and zero sequence component of current and phase angle has zero amplitude. At 46 ms D-phase current increases and voltage decreases while other phase currents remain same. As a result, the fault impedance increases. The fault current decreases, voltage increases and impedance increases as the fault location increases away from bus-2. The zero sequence component current starts to increase and phase angle starts to decrease in case of forward zone. The fuzzy schemes for the protection of a 6-phase line are based on these changes in currents, voltages, impedances and phase angles under pre-fault and post-fault conditions. The simple flowchart for the proposed scheme is depicted in Fig. 6. It consists of three stages, including faulty section detection, fault type classification and fault location.

III. MFES SCHEMES Modeling

Although the fuzzy theory was first introduced by Zadeh in 1960s, its application to the protection of transmission lines is somewhat recent. The FES is a popular computing scheme based on the concepts of

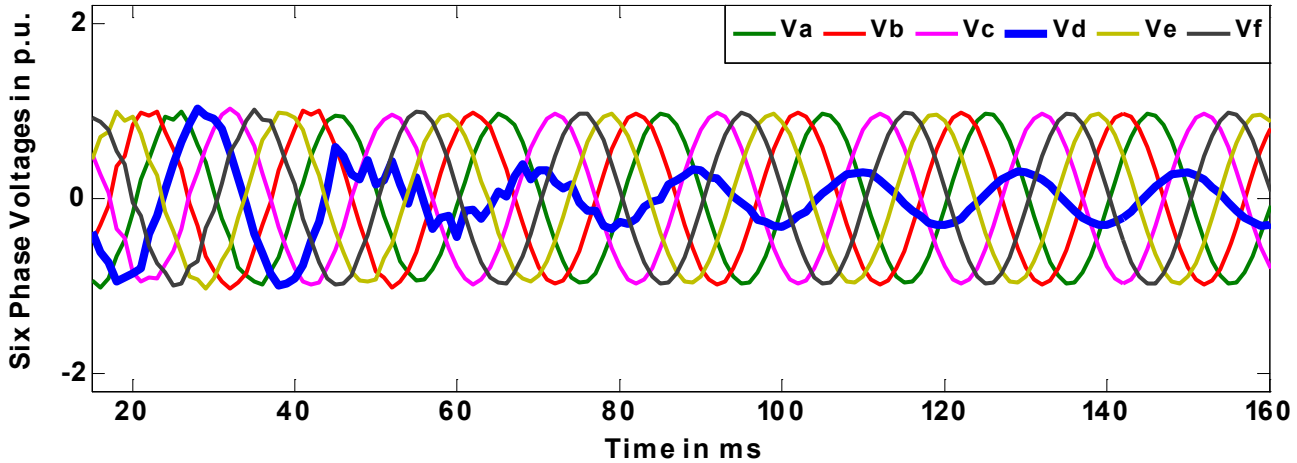


Fig. 2. Six-phase voltages during 'D-g' phase to ground fault.

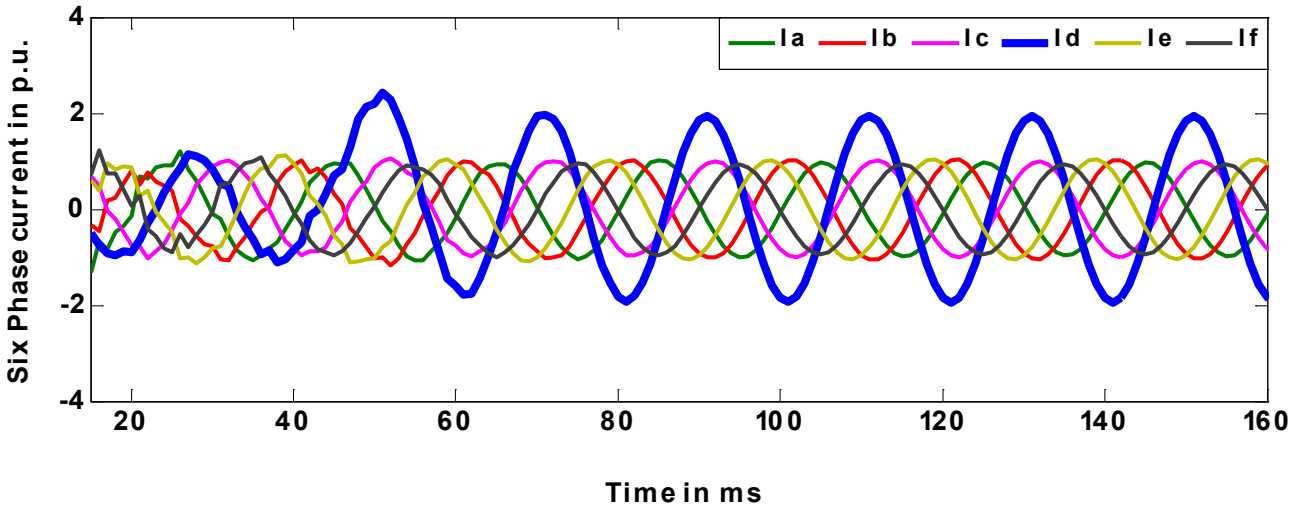


Fig. 3. Six-phase currents during 'D-g' phase to ground fault.

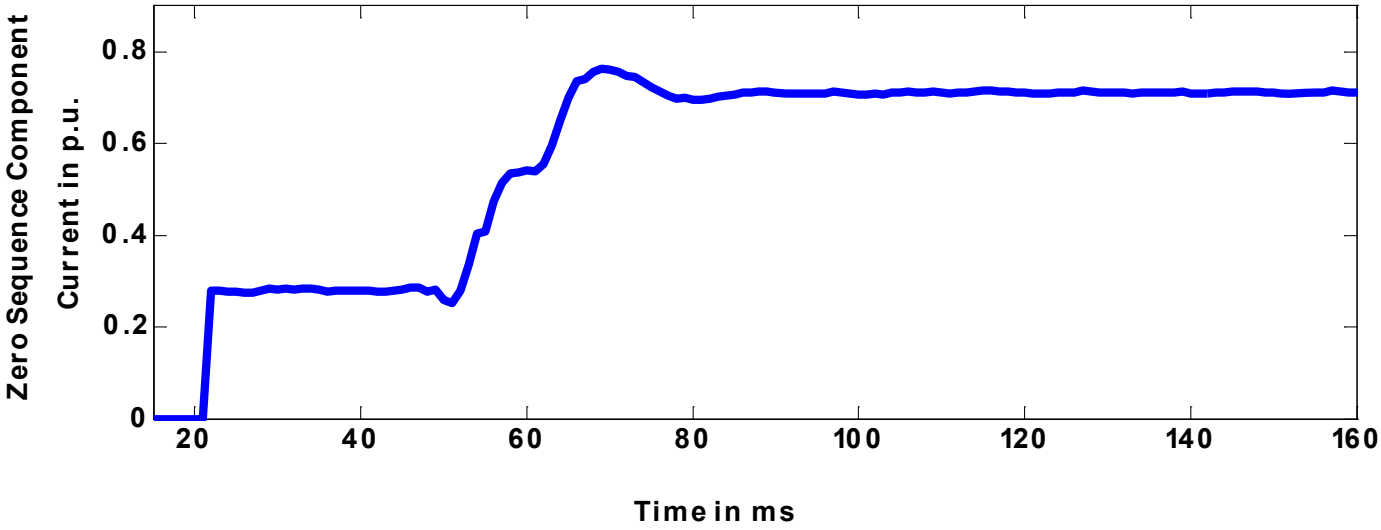


Fig. 4. Zero sequence current during 'D-g' phase to ground fault.

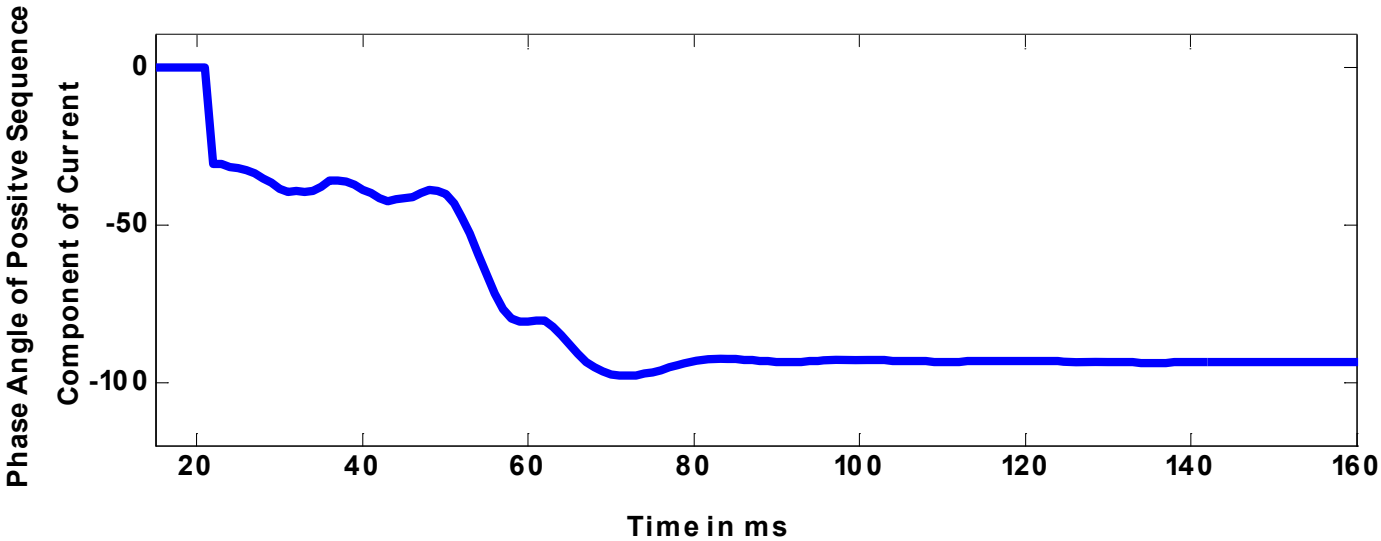


Fig. 5. Phase angle of positive sequence current during 'D-g' phase to ground fault.

fuzzy set theory, reasoning and if-then rules. It has been successfully employed in a wide variety of field's viz. data classification, automatic control, decision analysis, pattern recognition and robotics. The process of formulating the nonlinear mapping from a given input to an output based on fuzzy logic is known as fuzzy inference. Fig. 7 shows a diagram of general FES. The basic structure of FES consists of four conceptual components: fuzzification, rule base, inference engine and defuzzification. The fuzzification unit maps a vector input of scalar values into corresponding fuzzy sets. The rule base consists of linguistic rules that are often defined by domain experts. When a rule has multiple antecedents, the fuzzy operation is employed to determine a single firing strength in inference engine. The aggregator combines the output fuzzy sets into a single fuzzy set. The defuzzification unit then transfers the output fuzzy sets into a crisp value, which is the final output of FES. There are two kinds of FES schemes that can be implemented in the Fuzzy Toolbox: Mamdani and Takagi-Sugeno. Mamdani fuzzy expert system is the most commonly seen inference system. Mamdani fuzzy inference was first introduced by Mamdani and Assilian in 1975. The steps involved in the MFES scheme are explained in Fig. 8. The building of fuzzy schemes are described the next step.

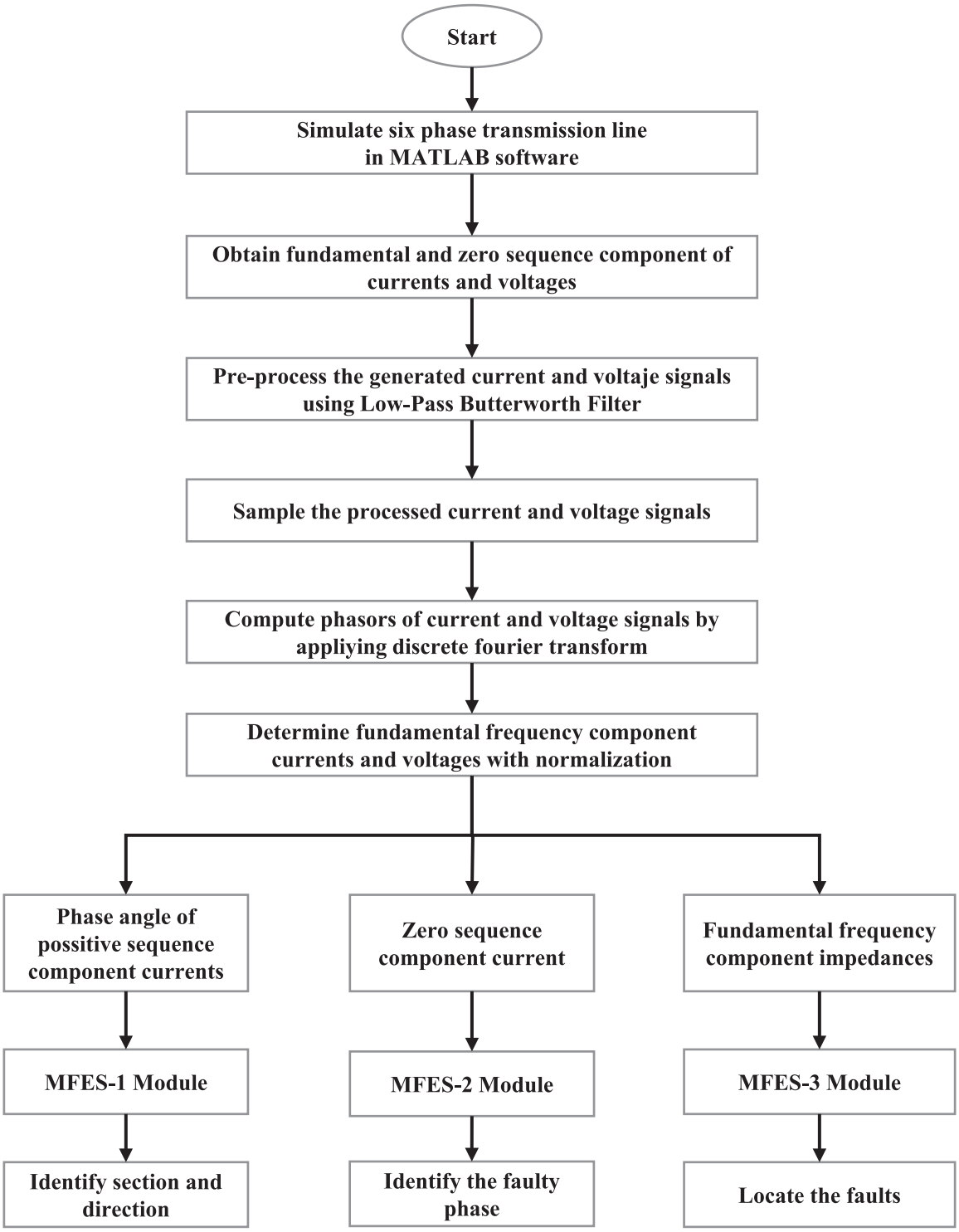


Fig. 6. Proposed protection scheme.

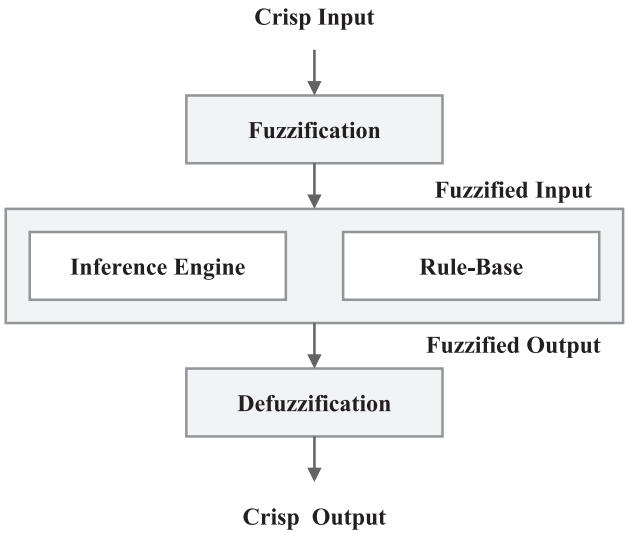


Fig. 7. Outline of FES.

Approach: MFES	
Stage 1:	***Fuzzification***
Step 1:	Determining a set of fuzzy rules
Step 2:	Fuzzifying the inputs using membership function
Stage 2:	***Rule Evaluation***
Step 1:	Combining the fuzzified inputs according to the fuzzy rules to establish rule strength.
Step 2:	Finding the consequence of the rule by combining the rule strength and the output MF
Step 3:	Combining the consequences to get an output distribution
Stage 3:	***Defuzzification***
Step 1:	Defuzzifying the output distribution

Fig. 8. Steps involved in the MFES scheme.

For the protection system three MFES schemes have been designed namely MFES-1, MFES-2 and MFES-3 to identify the faulty zone, classify and locate the faults, respectively. All the fuzzy schemes are modeled based on the Mamdani technique only. In the first scheme, MFES-1 is used by applying the phase angle of positive sequence current only to identify the faulty zone/section, whether the fault is on section-1 or on section-2. After that MFES-2 is developed by employing the fundamental component and zero sequence components of current to classify the faults. Further, MFES-3 is designed by using impedances in each phase to locate the fault. The first scheme (MFES-1) has one input parameter, 3 rules and one output. The second scheme (MFES-2) has seven input parameters, 120 rules and seven outputs whereas the third model (MFES-3) has also six input parameters, 8 rules and one output. The linguistic variable names, membership function (MF) names and MF type in each input-output for MFES-1, MFES-2 and MFES-3 are illustrated in Table-I. Fuzzy operators 'OR' and 'AND' have been recommended to connect input MFs. The minimum method is chosen to determine the firing strength. The aggregation used here is the maximum method. Then the centroid method of defuzzification is employed. After completion of modeling, proposed MFES schemes can provide accurate results by interring/ giving the proper input parameters.

IF-THEN rules for MFES-1

R1: IF (Phase angle is Small) THEN (Zone is Forward Zone)

R2: IF (Phase angle is Normal) THEN (Zone is No-Fault)

R3: IF (Phase angle is Large) THEN (Zone is Reverse Zone)

IF-THEN rules for MFES-2

IF (Input MF is Low) THEN (Output MF is Healthy)

IF (Input MF is High) THEN (Output MF is Faulty)

IF-THEN rules for MFES-3

R1: IF (FCZa is Z1) or (FCZb is Z1) or (FCZc is Z1) or (FCZd is Z1) or (FCZe is Z1) or (FCZf is Z1) THEN (L is FL1)

R2: IF (FCZa is Z2) or (FCZb is Z2) or (FCZc is Z2) or (FCZd is Z2) or (FCZe is Z2) or (FCZf is Z2) THEN (L is FL2)

R3: IF (FCZa is Z3) or (FCZb is Z3) or (FCZc is Z3) or (FCZd is Z3) or (FCZe is Z3) or (FCZf is Z3) THEN (L is FL3)

R4: IF (FCZa is Z4) or (FCZb is Z4) or (FCZc is Z4) or (FCZd is Z4) or (FCZe is Z4) or (FCZf is Z4) THEN (L is FL4)

R5: IF (FCZa is Z5) or (FCZb is Z5) or (FCZc is Z5) or (FCZd is Z5) or (FCZe is Z5) or (FCZf is Z5) THEN (L is FL5)

R6: IF (FCZa is Z6) or (FCZb is Z6) or (FCZc is Z6) or (FCZd is Z6) or (FCZe is Z6) or (FCZf is Z6) THEN (L is FL6)

R7: IF (FCZa is Z7) or (FCZb is Z7) or (FCZc is Z7) or (FCZd is Z7) or (FCZe is Z7) or (FCZf is Z7) THEN (L is FL7)

R8: IF (FCZa is Z8) or (FCZb is Z8) or (FCZc is Z8) or (FCZd is Z8) or (FCZe is Z8) or (FCZf is Z8) THEN (L is FL8)

A. MFES Modeling in MATLAB Software

1. Fuzzification

The transformation from input numerical to fuzzy is called as fuzzification. Numerical inputs are changed to their fuzzy equivalent by a family of MFs. In MFES-1, input represents phase angle. The output represents Zone. The three types of triangular MFs viz. Small, Normal and Large are designed for input as illustrated in Fig. 9 and three types of triangular MFs viz. Reverse Zone, No-Fault and Forward Zone are designed for output as illustrated in Fig. 10. In MFES-2, inputs represent FCIa, FCIb, FCIc, FCId, FCIe, FCIf and ZSCC. The outputs represent A, B, C, D, E, F, G. The two types of trapezoidal MFs viz. Low and High are designed for each input as illustrated in Fig. 11 and two types of triangular MFs viz. Faulty and Healthy are designed for each output as illustrated in Fig. 12. In MFES-3, inputs represent FCZa, FCZb, FCZc, FCZd, FCZe, FCZf. The output represents L. The eight types of triangular MFs viz. Z1, Z2, Z3, Z4, Z5, Z6, Z7 and Z8 are designed for each input as illustrated in Fig. 13 and eight types of triangular MFs viz. FL1, FL2, FL3, FL4, FL5, FL6, FL7 and FL8 are designed for each output as illustrated in Fig. 14.

TABLE I Input and Output Fuzzy Sets for Expert Systems

Schemes	System Variable	Linguistic Variable	MF Names	MF Type
MFES-1	Inputs (1)	Phase Angle	Small, Normal and Large	Triangular
	Output (1)	Zone	Reverse Zone, No-Fault and Forward Zone	Triangular
MFES-2	Inputs (7)	FCIa, FCIb, FCIc, FCId, FCIe, FCIf and ZSCC	Low and High	Trapezoidal
	Output (7)	A, B,C, D, E,F,G	Faulty and Healthy	Triangular
MFES-3	Inputs (6)	FCZa, FCZb, FCZc, FCZd, FCZe, FCZf,	Z1, Z2, Z3, Z4, Z5, Z6, Z7 and Z8	Triangular
	Output (1)	L	FL1, FL2, FL3, FL4, FL5, FL6, FL7 and FL8	Triangular

2. Fuzzy Inference Engine

After making input output fuzzy sets, fuzzy rules are constructed to evaluate faulty zone detection, fault type classification and fault location. The OR operator is performed with min operation and AND operator is performed with max operation. The fuzzy rules are framed in the form of IF-THEN based on the conditions and conclusions. The combining result of each rule is evaluated by fuzzy set operations. This mechanism is known as inference engine.

3. Defuzzification

The aggregation is changed into a numerical value for outputs. It is performed by the defuzzification technique. A single number indicates the fuzzy sets outcome. Centroid technique is employed to change the finalized fuzzy conclusions into numerical values. Fig. 15 shows centroid defuzzification for MFES-1 during reverse fault in MATLAB fuzzy software. Fig. 16 shows centroid defuzzification for MFES-2 during 'Ag' fault in MATLAB fuzzy software. Fig. 17 shows centroid defuzzification for MFES-3 during fault at 65 km in MATLAB fuzzy software.

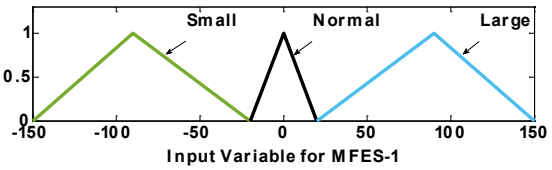


Fig. 9. Degree of MF for input variable of MFES-1.

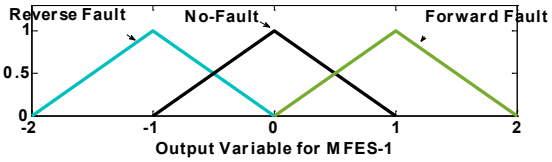


Fig. 10. Degree of MF for output variable of MFES-1.

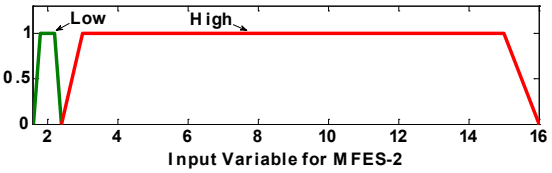


Fig. 11. Degree of MF for each input variable of MFES-2

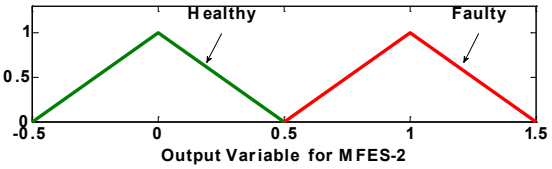


Fig. 12. Degree of MF for each output variable of MFES-2.

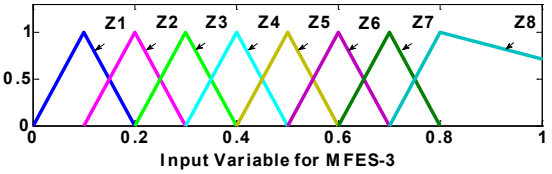


Fig. 13. Degree of MF for each input variable of MFES-3.

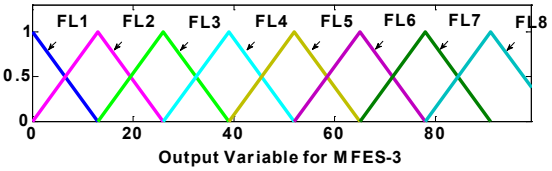


Fig. 14. Degree of MF for output variable of MFES-3.

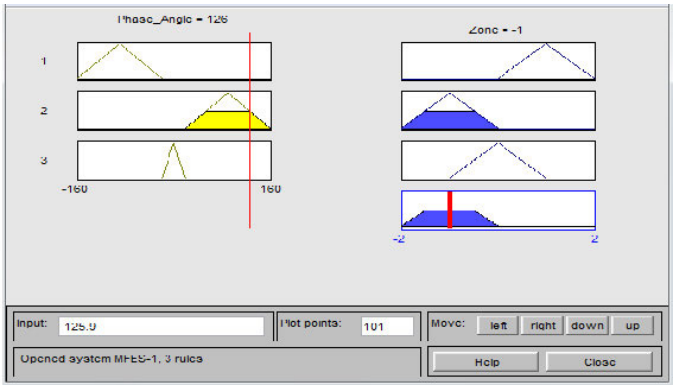


Fig. 15. Defuzzification for MFES-1.

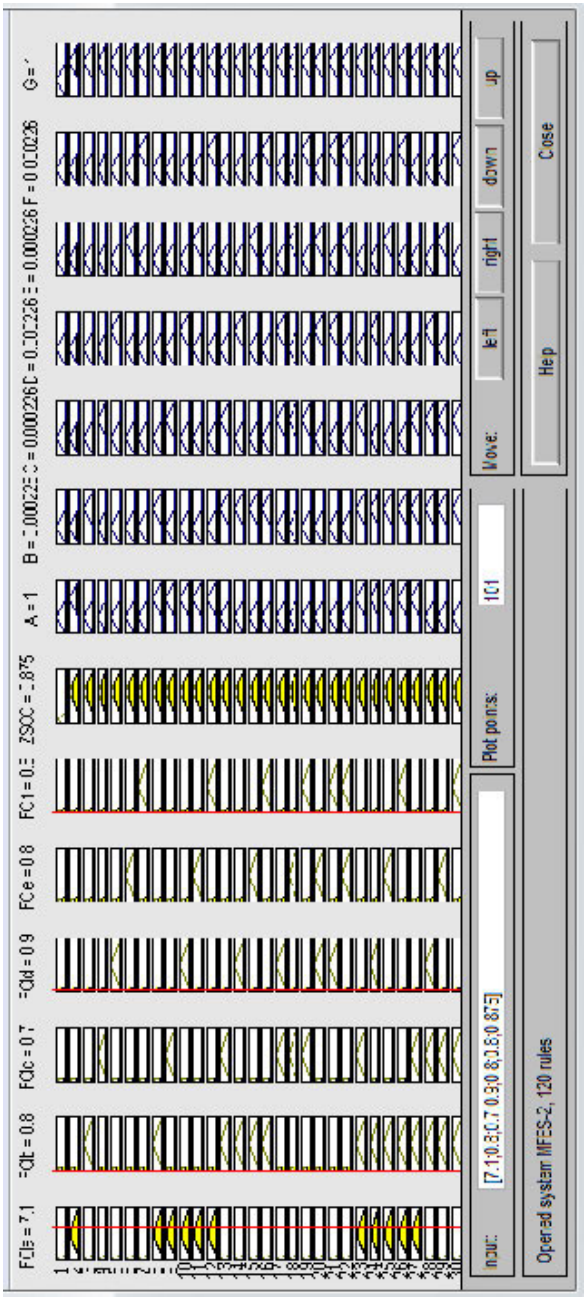


Fig. 16. Defuzzification for MFES-2.

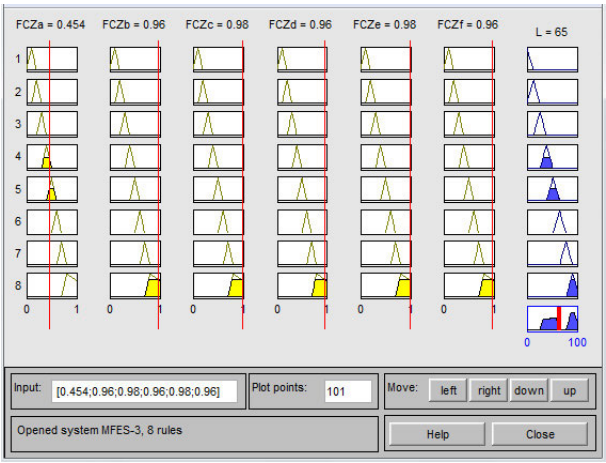


Fig. 17. Defuzzification for MFES-3.

IV. Results and Discussions

Proposed fuzzy based fault direction relaying, classification and location schemes are tested for different fault cases. The change of parameters considering fault types (forward and revere faults), fault inception angles, fault locations (including near end and far end) and fault resistances. The total number of test fault cases employed for testing the proposed fuzzy based scheme is 10,000 cases. The performance of MFES based schemes and comparison study is discussed in the following subsections.

A. The Performance of Directional Relaying, Fault Classifier and Fault Locator

Prior to classification, directional relaying and location estimation has been carried out. The fault direction relaying scheme traces the presence of faults and also determines the direction of faults. When there is no fault the response will be zero. If any fault occurs, the response starts changing to either -1 or +1 based on the reverse or forward faults, respectively. The simulation results of direction relaying detector and locator for various types of fault with changing fault location in reverse or forward direction from bus-2 are given in Table II. The simulation results of direction relaying detector and locator for various types of fault with changing fault resistance in reverse or forward direction from bus-2 are given in Table III. The simulation results of direction relaying detector and locator for various types of fault with changing fault inception angle in reverse or forward direction from bus-2 are given in Table IV. It is confirmed that most of the test samples have identified zone and located the faults.

The response of the proposed scheme is also tested for faults occurring in the boundary locations, i.e., near end and far end/remote faults. The traditional distance relaying technique reach setting is up to 85–95% of total length of line. Thus it is difficult to find the faults that occur between 95% and 100% of the length. Considering the objective of enhancing the reach setting value, the proposed scheme is developed. The response of the scheme for different far end/remote faults has been investigated by changing the fault location between 97 km and 99 km in each zone in steps of 1 km. The near-end faults occurring very close to bus-2 location in both the zones (within 3 % of the line) and the voltages at the relay will be very small. Therefore, the near end faults are challenging for the traditional impedance relay. The response of the scheme for different near end faults has been investigated by changing the fault location between 1 km and 3 km in each zone in steps of 1 km. The simulation results of different fault types at boundary locations are outlined in Table V. Thus the proposed scheme can protect total transmission line length and reach value of the line is 99.7%. The maximum absolute error percentage was 0.54% under phase-to-phase and phase-to-ground faults (including reverse, forward, near end and remote end faults. The % error in estimated fault location is calculated using (3).

$$\% \text{ Error} = \frac{L - Lo}{Line\ Length} \tag{3}$$

The simulation results of MFES-2 corresponding to various fault scenarios with change in fault parameters such as fault type, fault location, fault resistance and fault inception angle are listed in Table VI. The proposed MFES-2 scheme classified the faults correctly. It can be confirmed that the response time is within one cycle of post fault for all the tested fault scenarios.

TABLE II. Performance of Direction Relaying and Location Schemes in Case Of Different Fault Locations

Fault Zone	Type	L	MFES-1	MFES-3 (Lo)(km)	% Error
Reverse	Ag	5	-1	5.211	0.106
	ABg	9	-1	8.902	-0.049
	ABCg	18	-1	18.425	0.213
	ABCDg	26	-1	26.172	0.086
	ABCDEg	32	-1	31.833	-0.084
	ABCDEFg	41	-1	41.236	0.118
	AB	56	-1	56.183	0.092
	ABC	67	-1	67.266	0.133
	ABCD	74	-1	74.246	0.123
	ABCDE	83	-1	83.124	0.062
	ABCDEF	95	-1	94.842	-0.079
Forward	Ag	6	1	6.263	0.132
	ABg	15	1	14.936	-0.032
	ABCg	24	1	24.480	0.240
	ABCDg	38	1	37.812	-0.094
	ABCDEg	45	1	44.655	-0.173
	ABCDEFg	59	1	59.234	0.117
	AB	63	1	62.580	-0.210
	ABC	72	1	72.004	0.002
	ABCD	81	1	80.691	-0.155
	ABCDE	89	1	89.374	0.187
	ABCDEF	94	1	94.526	0.263

TABLE III. Performance of Direction Relaying and Location Schemes in Case of Different Fault Resistances at 33 Km Location

Fault Zone	Type	R (Ω)	MFES-1	MFES-3 (Lo)(km)	% Error
Reverse	Ag	110	-1	33.373	0.187
	ABg	90	-1	33.514	0.257
	ABCg	70	-1	33.439	0.220
	ABCDg	50	-1	33.262	0.131
	ABCDEg	30	-1	32.836	-0.082
	ABCDEFg	10	-1	33.328	0.164
	AB	56	-1	33.863	0.432
	ABC	67	-1	33.699	0.350
	ABCD	74	-1	33.218	0.109
	ABCDE	83	-1	32.941	-0.030
	ABCDEF	95	-1	33.360	0.180
Forward	Ag	110	1	33.252	0.126
	ABg	90	1	32.960	-0.020
	ABCg	70	1	32.981	-0.010
	ABCDg	50	1	32.693	-0.154
	ABCDEg	30	1	34.002	0.501
	ABCDEFg	10	1	32.340	-0.330
	AB	63	1	33.458	0.229
	ABC	72	1	33.024	0.012
	ABCD	81	1	33.698	0.349
	ABCDE	89	1	32.756	-0.122
	ABCDEF	94	1	33.215	0.108

TABLE IV. Performance of Direction Relaying and Location Schemes in Case Of Different Fault Inception Angles at 55 Km Location

Fault Zone	Type	ϕ (o)	MFES-1	MFES-3 (Lo)(km)	% Error
Reverse	Ag	30	-1	54.698	-0.151
	ABg	60	-1	55. 374	0.187
	ABCg	90	-1	54.624	-0.188
	ABCDg	120	-1	55.252	0.126
	ABCDEg	150	-1	55.490	0.245
	ABCDEFg	180	-1	55.604	0.302
	AB	210	-1	55.188	0.094
	ABC	240	-1	54.785	-0.108
	ABCD	270	-1	55.331	0.166
	ABCDE	300	-1	55.384	0.192
	ABCDEF	330	-1	54.792	-0.104
Forward	Ag	30	1	55.048	0.024
	ABg	60	1	55.434	0.217
	ABCg	90	1	55.266	0.133
	ABCDg	120	1	54.869	-0.066
	ABCDEg	150	1	56.008	0.504
	ABCDEFg	180	1	55.432	0.216
	AB	210	1	54.924	-0.038
	ABC	240	1	55.182	0.091
	ABCD	270	1	55.160	0.080
	ABCDE	300	1	54.988	-0.006
	ABCDEF	330	1	55.564	0.282

TABLE V. Performance of Direction Relaying and Location Schemes in Case of Different Near End Remote End Faults

Fault Zone	Type	L	MFES-1	MFES-3 (Lo)(km)	% Error
Near End Faults	Ag	1 km in zone-1	-1	1.334	0.167
	ABg	1 km in zone-2	1	1.253	0.127
	ABCg	1 km in zone-1	-1	0.680	-0.160
	ABCDg	2 km in zone-1	-1	2.096	0.048
	ABCDEg	1 km in zone-1	-1	1.532	0.266
	ABCDEFg	3 km in zone-2	1	3.432	0.216
	AB	1 km in zone-1	-1	1.286	0.143
	ABC	3 km in zone-2	1	2.645	-0.178
	ABCD	2 km in zone-1	-1	2.021	0.011
	ABCDE	1 km in zone-2	1	1.540	0.270
	ABCDEF	2 km in zone-1	-1	2.460	0.230
Far End Faults	Ag	99 km in zone-2	1	99.122	0.061
	ABg	98 km in zone-2	1	97.890	-0.055
	ABCg	97 km in zone-2	1	97.462	0.231
	ABCDg	99 km in zone-1	-1	99.260	0.130
	ABCDEg	98 km in zone-2	1	97.910	-0.045
	ABCDEFg	97 km in zone-2	1	97.454	0.227
	AB	99 km in zone-2	1	99.180	0.090
	ABC	98 km in zone-1	-1	98.500	0.250
	ABCD	97 km in zone-2	1	97.422	0.211
	ABCDE	99 km in zone-1	-1	98.646	-0.177
	ABCDEF	98 km in zone-2	1	98.380	0.190

Fig. 18 explains the output of MFES-1 in which 'Zone' goes -1 at 40 ms showing there is reverse fault. Fig. 19 explains the outputs of MFES-2 in which 'A and G' phases go 1 at 45 ms, with other outputs B, C D E and F remaining 0 (no fault) showing there is Ag-fault. Fig. 20 explains the output of MFES-3 in which 'L' goes 65 at 40 ms showing there is a fault occurring at 65 km. Following directional relaying, classification and fault location estimation, the comparison with other techniques are discussed in the next subsection.

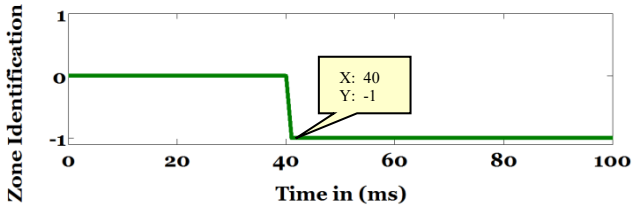


Fig. 18. Output of MFES-1 during reverse fault at 40 ms.

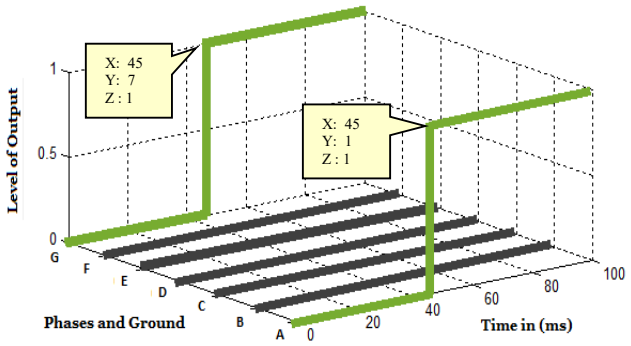


Fig. 19. Output of MFES-2 during Ag fault at 40 ms.

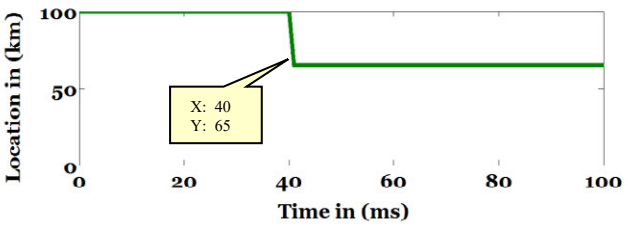


Fig. 20. Output of MFES-3 during fault at 40 ms at 65 km.

B. Comparison with Other Techniques

To further judge the expediency of the scheme, the comparison analysis has been carried out for reach setting and scheme used. The proposed scheme has been compared with the other ones reported in papers listed in Table VII. Most of the existing schemes in six phase line [4]-[8] are non-directional relays. The proposed relay is directional and deciphered the difficulty of distinguishing between reverse and forward faults and providing the precise fault location in six-phase lines. ANN based fault location [7], directional relying [15] and classification [5] techniques are developed. However, the drawback of ANN techniques is that it requires large data samples set to learn the relationship between fault patterns with changing fault parameters and no-fault which needs tedious training process. Also, the choice of the training algorithms, mean square error goal, number of epochs, hidden layers, activation functions and number of neurons is a difficult task. Thus, a technique without complex computation and without training will be more advantageous, like the proposed fuzzy technique. The reach setting of line length in [11] is 99.5%. The test results of the proposed method in the present paper; the reach setting is 99.7% without training effort. A fuzzy technique [24] is addressed for 3-phase line and is also directional relay, but its reach is up to 99% and it does not deal with a six phase line. The proposed scheme not only gives the minimum error but also attains the same with better reach setting in six phase line. Simulations confirmed high efficacy and satisfactory performance of the proposed directional relaying, classification and fault location scheme compared with other reported approaches. Thus, the fuzzy relay is a better option for direction relaying, classification and fault location compared to all other techniques in six phase line.

V. Conclusion

This study dealt with performance improvement of zone identification and fault location in six phase line. The fault protection is achieved through the MFES scheme, which works on phase angle, currents and impedance samples acquired at single terminal of line. The results show that the proposed MFES based scheme can be effectively applied for zone identification of six phase line. Simulation results show that the proposed relays could give protection to both the sections

TABLE VI. Performance of Fault Classifier in Case Of Different Fault Locations, Resistances and Inception Angles

Parameter Varied	Fault Type	L (km)	R (Ω)	ϕ (o)	Time (ms)	MFES-2 A	B	C	D	E	F	G
L	Ag	4	50	45	2	1	0	0	0	0	0	1
	ABg	15	50	45	6	1	1	0	0	0	0	1
	ABCg	23	50	45	5	1	1	1	0	0	0	1
	ABCDg	29	50	45	2	1	1	1	1	0	0	1
	ABCDEg	38	50	45	7	1	1	1	1	1	0	1
	ABCDEFg	46	50	45	7	1	1	1	1	1	1	1
	ABg	53	0	45	3	1	1	0	0	0	0	0
	ABCg	61	0	45	2	1	1	1	0	0	0	0
	ABCDg	79	0	45	4	1	1	1	1	0	0	0
	ABCDEg	88	0	45	3	1	1	1	1	1	0	0
	ABCDEFg	94	0	45	2	1	1	1	1	1	1	0
R	Ag	41	110	80	7	1	0	0	0	0	0	1
	ABg	41	90	80	2	1	1	0	0	0	0	1
	ABCg	41	70	80	1	1	1	1	0	0	0	1
	ABCDg	41	50	80	4	1	1	1	1	0	0	1
	ABCDEg	41	30	80	3	1	1	1	1	1	0	1
	ABCDEFg	41	10	80	3	1	1	1	1	1	1	1
ϕ	Ag	12	100	30	5	1	0	0	0	0	0	1
	ABg	12	100	60	1	1	1	0	0	0	0	1
	ABCg	12	100	90	3	1	1	1	0	0	0	1
	ABCDg	12	100	120	2	1	1	1	1	0	0	1
	ABCDEg	12	100	150	2	1	1	1	1	1	0	1
	ABCDEFg	12	100	180	5	1	1	1	1	1	1	1
	ABg	12	0	210	7	1	1	0	0	0	0	0
	ABCg	12	0	240	4	1	1	1	0	0	0	0
	ABCDg	12	0	270	2	1	1	1	1	0	0	0
	ABCDEg	12	0	300	2	1	1	1	1	1	0	0
	ABCDEFg	12	0	330	1	1	1	1	1	1	1	0

TABLE VII. Comparison with Other Existing Works

Ref. No.	Proposed Scheme	Protection Type	Reach Setting	% Error	Transmission Line
[4]	ANN	Classification	-	-	6-Phase
[5]	ANN	Classification	-	-	6-Phase
[6]	ANN	Classification and Location	-	0.41	6-Phase
[7]	ANN	Location	-	0.73	6-Phase
[8]	ANN	Classification and Location	-	0.688	6-Phase
[11]	Modular Neuro-Wavelet	Zone Identification and fault Location	99.5	0.675	6-Phase
[15]	ANN	Directional relaying	99.5	-	3-Phase
[24]	Fuzzy Inference System	Directional Relaying, Classification and Location	95	2	3-Phase
Proposed method	Fuzzy Inference System	Directional relaying, Classification and Location	99.7	0.504	6-Phase

of the six phase line. Neither fault inception angle information nor fault type is a prerequisite for this scheme. The schemes have the advantage of automatically tuning the MFs for an optimal design. The proposed directional relaying offers protection up to 99.7% of the line in both reverse zone and primary zone. Even though the proposed scheme is simple and easy, the improvement is significant. .

References

[1] James, R. Stewart, Thomas F. Dorazio and Matthew T. Brown, "Insulation Coordination, Environmental and System Analysis of Existing Double Circuit Line Reconfigured to Six-Phase Operation," *IEEE Transactions on Power Delivery*, vol.7, no.3, pp. 1628-1633, 1992.

[2] S.S. Venkata, W.C. Guyter, J.Kondragunta and N.B. Butt, "EPPC - A Computer Program for Six-Phase Transmission Line Design," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-101, no. 7, pp. 1859-1869, 1982.

[3] Laurie Oppel Edw ard Krizauskas, "Evaluation of the Performance of Line Protection Schemes on the NYSEG Six Phase Transmission System," *IEEE Transactions on Power Delivery,* vol.14, no.1, pp. 110-115, 1990.

[4] Ebha Koley, Anamika Jain, A.S.Thoke, Abhinav Jain and Subhojit Ghosh, "Detection and Classification of Faults on Six Phase Transmission Line using ANN," *IEEE International Conference on Computer & Communication Technology,* MNNIT, Allahabad, pp. 100-103, 2011.

[5] Ravi Kumar, Ebha Koley, Anamika Yadav and A.S. Thoke, "Fault Classification of Phase to Phase Fault in Six Phase Transmission Line using Haar Wavelet and ANN," *in Proceedings of IEEE International Conference on Signal Processing and Integrated Network.,* Noida, India, pp. 5-8, 2014.

[6] A. Naresh Kumar and Chakravarthy M, "Simultaneous Fault Classification and Localization Scheme in Six Phase Transmission Line Using Artificial Neural Networks," *Journal of Advanced Research in Dynamical & Control Systems,* vol.10, no.3, pp. 342-349, 2018.

[7] Ebha Koley, Anamika Yadav and Aniruddha Santosh Thoke, "A New Single-Ended Artificial Neural Network-based Protection Scheme for Shunt Faults in Six-Phase Transmission Line," *International Transactions on Electrical Energy Systems,* vol.25, no.7, pp. 1257-1280, 2015.

[8] Ebha Koley, Khushaboo Verma and Subhojit Ghosh, "An Improved Fault Detection Classification and Fault Location Scheme based on Wavelet Transform and Artificial Neural Networks for Six Phase Transmission Line using Single End Data Only," *Springer Plus*, 2015.

[9] Apostolov AP and George W, "Protecting NYSEG's Six-Phase Transmission Line," *IEEE computer applications in power,* vol.5, no.4, pp. 33-36, 1992.

[10] Apostolov AP and Raffensperger RG (1996), "Relay Protection Operation for Faults on NYSEG's Six Phase Transmission Line," *IEEE Trans Power Delivery,* vol.11, no.1, pp. 191-196.

[11] Ebha Koley, Khushaboo Verma and Subhojit Ghosh, "A Modular Neuro-Wavelet Based Non-Unit Protection Scheme for Zone Identification and Fault Location in Six Phase Transmission Line," *Neural Computing and Applications,* vol.28, no.6, pp. 1369-1385, 2017.

[12] Anamika Yadav and Aleena Swetapadma, "A Novel Transmission Line Relaying Scheme for Fault Detection And Classification using Wavelet Transform and Linear Discriminant Analysis," *Ain Shams Engineering Journal,* vol.6, pp. 199-209, 2015.

[13] Aleena Swetapadma and Anamika Yadav, "High-Speed Directional Relaying using Adaptive Neuro-Fuzzy Inference System and Fundamental Component of Currents," *IEEJ Transactions on Electrical and Electronic Engineering,* vol.10, pp. 653-663, 2015.

[14] S.R. Samantaray, "Decision Tree-based Fault Zone Identification and Fault Classification in Flexible AC Transmissions-based Transmission Line," *IET Generation, Transmission & Distribution*, vol.3, no.5, pp. 425-436, 2009.

[15] Anamika Yadav, Yajnaseni Dash and V. Ashok, "ANN based directional Relaying Scheme for Protection of Korba-Bhilai Transmission Line of Chhattisgarh State," *Protection and Control of Modern Power Systems,* no.1, 2016.

[16] García, C. G., E. R. Nunez-Valdez, V. García-Díaz, C. Pelayo G-Bustelo and J. M. Cueva-Lovelle, "A Review of Artificial Intelligence in the Internet of Things," International Journal of Interactive Multimedia and Artificial Intelligence, vol.5, no.4, pp. 9-20, 2019.

[17] Benamina, M., B. Atmani, and S. Benbelkacem, "Diabetes Diagnosis by Case-Based Reasoning and Fuzzy Logic," International Journal of Interactive Multimedia and Artificial Intelligence, vol.5, no.3, pp. 72-80, 2018.

[18] Farhane, N., I. Boumhidi, and J. Boumhidi, "Smart Algorithms to Control a Variable Speed Wind Turbine," International Journal of Interactive Multimedia and Artificial Intelligence, vol.4, no.6, pp. 88-95, 2017.

[19] Praveen Kumar Mishra and Anamika Yadav, "A Single Ended Fuzzy Based Directional Relaying Scheme For Transmission Line Compensated by Fixed Series Capacitor," ISDA, VIT Vellore, pp. 6-8, 2018.

[20] Aleena Swetapadma and Anamika Yadav, "Fuzzy Inference System Approach for Locating Series, Shunt, and Simultaneous Series-Shunt Faults in Double Circuit Transmission Lines," Computational Intelligence and Neuroscience, vol. 2015, Article ID 620360, 12 pages, 2015. https://doi.org/10.1155/2015/620360.

[21] A. Naresh Kumar and Chakravarthy. M, "Fuzzy Inference System Based Distance Estimation Approach for Multi Location and Transforming Phase to Ground Faults in Six Phase Transmission Line," International Journal of Computational Intelligence Systems, vol.11, no.1, pp. 757-769, 2018.

[22] A. Naresh Kumar, Ch Sanjay and M Chakravarthy "Fuzzy Expert System based Protection for Double Circuit Incomplete Journey Transmission Lines," International Journal of Computational Intelligence & IoT, vol.2, no.1, pp. 351-355, 2019.

[23] A. Naresh Kumar, Ch Sanjay and M Chakravarthy, "Fuzzy Inference System-based Solution to Locate the Cross-Country Faults in Parallel Transmission Line," International Journal of Electrical Engineering & Education, 2019.

[24] Anamika Yadav and Aleena Swetapadma, "Enhancing the performance of transmission line directional relaying, fault classification and fault location schemes using fuzzy inference systems," IET Generation, Transmission and Distribution, vol.9, no.6, pp. 580-591, 2015.

A. Naresh Kumar

A Naresh Kumar received the B. Tech. degree in Electrical and Electronics Engineering from Jawaharlal Nehru Technological University, Jagitiala, India in 2011. He received his M.Tech degree in Electrical Engineering from National Institute of Technology, Raipur, India, in 2013. Presently he is pursuing his Ph.D. from department of Electrical and Electronics Engineering, GITAM University, Hyderabad. His areas of interest are power system protection, smart grid and soft computing applications.

Ch. Sanjay

Prof. Ch. Sanjay (UG/PG/Ph.D from reputed Government Universities in India) had joined GITAM University, Hyderabad as a Distinguished Professor and Founder Director, from June, 2009 till 2014, currently working as Principal and Dean, School of Technology, and has been a member of Board of Management, IQAC and Academic Council since 2009. He was Keynote Speaker for 64 International/ National Conference and Co-Chair for 8 International Conferences held abroad. He is Member, Editorial Board/ Reviewer for more than 20 international journals and evaluated 19 Doctorial theses in Mechanical / Manufacturing / Management / Information Technology and presently guiding 10 Research Scholars. To his credit, Prof. Sanjay has 4 books published by reputed publishers and authored more than 60 Research papers in Peer Reviewed, Indexed Journals and Conference Proceedings and completed 6 international research projects. Recipient of many International and National Awards such as Academic Excellence, Research Excellence, Best Engineer, Best Principal, Best Academic Administrator, Education Leadership and Engineers Educator award.

M. Chakravarthy

Dr. M Chakravarthy received his Phd degree in Electrical and Electronics Engineering from Jawaharlal Nehru Technological University, Hyderabad, India in 2013. He is Professor in the department of Electrical and Electronics Engineering, Vasavi College of Engineering, Hyderabad, India. His research areas are smart grid, DC-DC converters, automation and industrial control. He is a member of IEEE.

Learning Models for Semantic Classification of Insufficient Plantar Pressure Images

Yao Wu[1], Qun Wu[2, 3, 4]*, Nilanjan Dey[5], R. Simon Sherratt[6]

[1] Department of Industrial Design, Wenzhou Business College, Wenzhou (PR China)
[2] Institute of Universal Design, Zhejiang Sci-Tech University, Hanzhou (PR China)
[3] Collaborative Innovation Center of Culture, Creative Design and Manufacturing Industry of China Academy of Art, Hangzhou (PR China)
[4] Zhejiang Provincial Key Laboratory of Integration of Healthy Smart Kitchen System, Hangzhou (PR China)
[5] Department of Information Technology, Techno India College of Technology, West Bengal (India)
[6] Department of Biomedical Engineering, the University of Reading, Berkshire (United Kingdom)

Received 20 December 2019 | Accepted 16 February 2020 | Published 1 March 2020

unir
LA UNIVERSIDAD EN INTERNET

Abstract

Establishing a reliable and stable model to predict a target by using insufficient labeled samples is feasible and effective, particularly, for a sensor-generated data-set. This paper has been inspired with insufficient data-set learning algorithms, such as metric-based, prototype networks and meta-learning, and therefore we propose an insufficient data-set transfer model learning method. Firstly, two basic models for transfer learning are introduced. A classification system and calculation criteria are then subsequently introduced. Secondly, a data-set of plantar pressure for comfort shoe design is acquired and preprocessed through foot scan system; and by using a pre-trained convolution neural network employing AlexNet and convolution neural network (CNN)-based transfer modeling, the classification accuracy of the plantar pressure images is over 93.5%. Finally, the proposed method has been compared to the current classifiers VGG, ResNet, AlexNet and pre-trained CNN. Also, our work is compared with known-scaling and shifting (SS) and unknown-plain slot (PS) partition methods on the public test databases: SUN, CUB, AWA1, AWA2, and aPY with indices of precision (tr, ts, H) and time (training and evaluation). The proposed method for the plantar pressure classification task shows high performance in most indices when comparing with other methods. The transfer learning-based method can be applied to other insufficient data-sets of sensor imaging fields.

Keywords

Machine Learning, Artificial Neural Networks, Feature Extraction, Image Processing, Image Analysis, Image Classification.

DOI: 10.9781/ijimai.2020.02.005

I. Introduction

DATA in our social society, scientific research, and other fields, are generating at an unprecedented rate and being widely collected and stored. To further exploring the information or knowledge hiding in the datasets, intelligent technologies-based data processing has been developed and becomes the consensus of the current academic and industrial circles. However, the lack of labeled data has become increasingly prominent with the accumulation of many new data-sets. While not every field will spend as much manual labeling to produce some data like ImageNet [1], supervised learning can solve many important problems. Transfer Learning (TL) plays a very important role for processing the unlabeled datasets. In traditional classification learning models, there are two basic assumptions to guarantee the accuracy and reliability of the classification model; the first is that how the training samples for learning to the new test samples satisfying the same distribution independently; the second is that there must be enough available training samples to learn a good classification model [2]–[3].

* Corresponding author.
E-mail address: wuq@zstu.edu.cn

In practice however, these two conditions are often unsatisfactory. Over time, the previously available labeled sample data may become unavailable resulting in semantic and distribution barriers with new testing samples. Reliable labeled sample data are often scarce and hard to be obtained. This resulted in a very important problem in Machine Learning (ML), how to use insufficient labeled samples or source domain data to predict target domains with different data distribution of reliable models. Recently, TL has attracted extensive attention and research [4]. Transfer learning uses existing knowledge to solve problems in different but related fields; it aims to solve the learning problem in the target region by transferring the existing knowledge, thus relaxing the two basic assumptions in traditional machine learning, while only a small amount of labeled sample data is available in the target region. The more factors shared by two different fields then the easier it will be. Otherwise, it will be more difficult, and even "negative transfer" will occur, which has side effects in practice [5].

Scholars have carried out extensive research on TL recently; and many of them have focused on different technologies to study TL algorithms. TL arises from the mining technology of incomplete or insufficient data-sets; and as one of the Deep Learning (DL) technologies, it has been used in various fields to solve problems. The target accuracy of more than 90% was easily achieved by

previous studies [6]– [7]. However, DL is a data-hungry technology that requires many annotated samples to work with. In practice, there are many problems due to the lack of labeled data, and the cost of obtaining labeled data is also very large, such as in the field of medical treatment and information security. Therefore, this inevitably leads to the well-known small sample problem, with new classes in the training process that have never been seen before. It can only use a few labeled samples of each class without changing the trained model [8]. Taking image classification data as an example, the traditional method is to obtain a model based on the training set, and then annotate the test set automatically. The small sample problem is that many classified data-sets can be processed using a few labeled data-sets. When the amount of tagged data-set is relatively small, these rare categories need to be generalized without additional training. Few-shot, one-shot [9]– [11], and Zero-Shot Learning (ZSL) models are widely used to address this issue. TL is highly related to the few-shot and one-shot learning models [12]. Abderrahmane et al. [13] introduced a zero-shot haptic recognition algorithm for robots in interacting with their environment. Ji et al. [14]-[15] introduced a Manifold-regularized Cross Modal Embedding (MCME) approach for ZSL. Combining ontology and reinforcement learning for Zero-Shot Classification (ZSC) [16]–[17], Kernelized Linear Discriminant Analysis (KLDA), Central-Loss based Network (CLN) and Kernelized Ridge Regression (KRR) for ZSL [18] has also been developed subsequently. Yu et al. [19] introduced a regularized cross-modality ranking (ReCMR) to capture the semantic information from heterogeneous sources by using intra/intermodals.

In ZSL, intuitive decision-making is more reliable based on costs and benefits [20]. Some remarkable classification results deployed plenty of new class by using fast prototype-based learning algorithms [21]-[24]. From a TL aspect, the distribution of labeled source domain data is generally different from that of unlabeled target domain data, so the labeled sample data is not necessarily useful [25]. Therefore, the selection of the training samples which are most beneficial to target domain classification is of utmost importance. The TL algorithms can be divided into those applied to single-source domains and those applied to multiple source domains. TL has been widely applied in text classification and clustering, emotional classification, image classification and collaborative filtering. Soudani and Barhoumi [26] extracted features from the convolutional part to improve the segmentation performance using two pre-trained architectures (VGG16 and ResNet50).

Plantar pressure test and gait analysis is an internationally advanced technology based on the principle of biomechanics to detect the structure of the lower limbs of the human body, to assess and predict future foot diseases, and to provide scientific rehabilitation methods. The reaction force of the ground can be divided into static and dynamic foot pressure, which respectively represent the ground reaction force that the human body receives when standing statically and dynamically walking and running and jumping. Most of the foot diseases are related to normal pressure changes in the soles of the feet, and the two usually affect each other. Therefore, understanding the distribution of normal foot pressure is not only an effective tool for the diagnosis and evaluation of foot diseases but also is an important guider in restoring the normal or near-normal distribution of the foot pressure and restoring the function of the foot.

The pressure distribution of the human foot can directly reflect the pressure value of the various parts of the foot when standing or running and can also indirectly reflect the structure and function of the foot and the control of the knee, hip, spine and even the entire body posture. Testing and analyzing the plantar pressure can obtain the pressure information of the normal and abnormal soles of the human body in various postures. Through the dynamic collection of the pressure data of the sole, the biomechanical properties of the plantar pressure can be determined for the early prediction, diagnosis and treatment of deformed foot, various types of foot diseases and diabetic foot ulcers; quantitative assessment of the degree of joint disease and postoperative efficacy in orthopedics and orthopedic surgery; Walking training and designing intelligent prosthetics and other fields. Therefore, a comprehensive understanding of the changes in plantar pressure facilitates the understanding of the biomechanics and function of normal feet, clinical medical diagnosis, disease degree determination, postoperative efficacy evaluation, rehabilitation research and the design of various types of orthopedic shoes and sports shoes, which are of great significance.

Deschamps et al. [27] proposed a high-resolution pixel-level analysis and 12 of the standardized foot-pressure barometers in the original experimental data-set (including 97 diabetic patients and 33 non-diabetic patients). The new cohort of diabetic patients determines the classification recognition rate and its sensitivity and specificity to assess the classification effect. Ramirez-Bautista et al. [28] used plantar pressure data to detect disease progression and the use of different electronic measurement systems for corresponding analysis suggests a hybrid algorithm for classifying plantar images. Sommerset et al. [29] combined other physiological signals to study the effects of plantar pressure on compressible atherosclerosis. Xia et al. [30] analyzed the background plantar pressure image (PPI) of high temporal and spatial resolution and experimental results perform higher effectiveness in terms of mean square error (MSE), exclusive OR (XOR) and mutual information (MI). Plantar pressure imaging is different from the general image. Pressure images are formed by discrete values of pressure. Colors are used to reflect different pressure and pressure values. These pressure values will be applied to the image intelligent classification method in this paper. Therefore, we screened the current popular deep learning method to process the image, thus transforming the data set of discrete pressure values into image processing problems. The research motivation of our work is to find an effective transfer learning method applied for such insufficient datasets, including lack of labelled datasets. The finding of the research is that we used the images forms of the sensor generated dataset for training and classification by using an improved convolutional neural network, and the significance of the network models performs high efficiency by comparing with VGG, ResNet, AlexNet and pre-trained CNN, etc., in the classification indices.

The structure of the rest of the paper is: Section 2 introduces the basic models for plantar pressure image classification, and improved models are described in Section 3. Section 4 deploys the results and discussion, while Section 5 includes concluding remarks and future works.

II. Methods

A. Finetune-Based Modeling

DL requires a large amount of data and computing resources and takes a lot of time to train the model, but it is difficult to meet these requirements in practice. The use of TL can effectively reduce the amount of data, calculations, and calculation time, and can be customized in the new business needs of the scenario. Transfer learning is not an algorithm but a machine learning idea, and its application to deep learning is fine-tune. By modifying the structure of the pre-trained network model (such as modifying the number of sample category outputs), selectively load the weight of the pre-trained network model (usually loading all the previous layers except the last fully connected layer, also called the bottleneck layer), and then use it retraining the model on the dataset is the basic step of fine-tuning. Fine-tuning can quickly train a model with relatively small amount of data achieving good results.

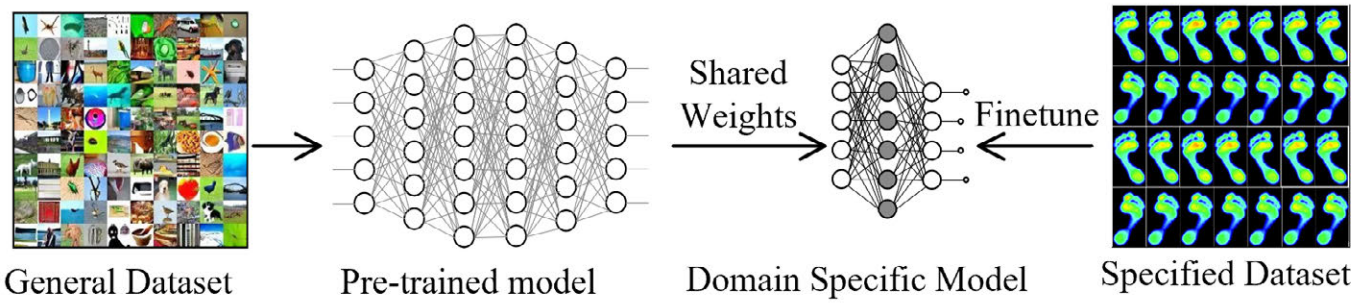


Fig. 1. Pre-trained convolutional neural network for shared weights finetune specified data-set using general image database- ImageNet.

Finetune-based models have been widely used in the field of DL. By obtaining a certain amount of tagging data, a basic network can be established for fine-tuning. This basic network is acquired through large-scale data sets with rich tags, such as ImageNet, or e-commerce data, known as universal data domains. Then, training is performed on specific data domains. While the parameters of the basic network will be fixed in the process, and the domain-specific network parameters will be trained, including to set the fixed layer and learning rate, etc. This model method can be relatively fast and does not significantly depend on the amount of data. Fig. 1 shows the framework of a finetune-based model for the plantar pressure image specified data-set.

B. Siamese Neural Networks

Siamese Neural Networks (SNN) measure how similar the two inputs are. The twin neural network has two inputs (Input1 and Input2), and the two input feeds into two neural networks (network1 and network2). These two neural networks map the inputs to the new space respectively, forming the input in the new space. Trough the calculation of Loss, the similarity of the two inputs is evaluated. This method restricts the input structure and automatically discovers features that can be generalized from new samples. Supervised metric learning based on twin networks is trained, and then one/few-shot learning is performed by reusing the features extracted from that network [31] – [34]. It is a two-way neural network. It combines samples of different classes into pairs for the network training. At the top level, it calculates loss through a distance cross-entropy. In forecasting, taking 5 way-5 shot as an example, five samples are randomly selected from five classes, and the data with 25 mini-batch is input into the network. Finally, 25 values are obtained. The category with the highest score is taken as the forecasting result, as shown in Fig. 2.

The network structure is shown in Fig. 3, it is an eight-layer deep convolution twin network. The graph shows only one of the calculations. After the 4096-dimensional full-connection layers of the network, the component-wise L1 distance calculation is performed to generate a 4096-dimensional eigenvector, a probability of 0 to 1 is obtained by sigmoidal activation as the result of whether two input samples are similar.

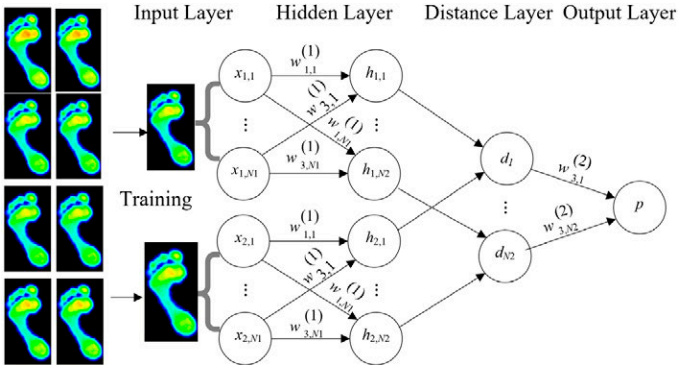


Fig. 2. Typical Siamese Neural Networks for plantar pressure image data-set training using distance layer and shared weights.

C. Matching Networks

Without changing the network model, matching networks can generate labels for unknown categories. Its main innovation lies in the process of modeling and training. For the innovation of the modeling process, a matching network based on memory and attention is proposed, which makes it possible to learn quickly. For the innovation of the training process, this work is based on a principle of traditional ML, that is training and testing are to be carried out under the same conditions [35]-[36]. It is proposed that the network should constantly look at the insufficient samples of each type during training, which will be consistent with the testing process. Specifically, it attempts to obtain a mapping from support set S (composed of K samples and tags) to classifier $\hat{Y}$, which is a network of $P(\hat{Y}|\hat{X},S)$. It gives the label $\hat{Y}$ for each unknown test sample $\hat{X}$ based on the current *S*, and the label maximizes *P*. This model is described as follows,

$$\arg\max_y \; P(Y|\hat{X},S) \tag{1}$$

$$\hat{Y} = \sum_{i=1}^{k} a(\hat{X},x_i)y_i \tag{2}$$

where, a is an attention (i.e. the distribution of classes on *S*) being a

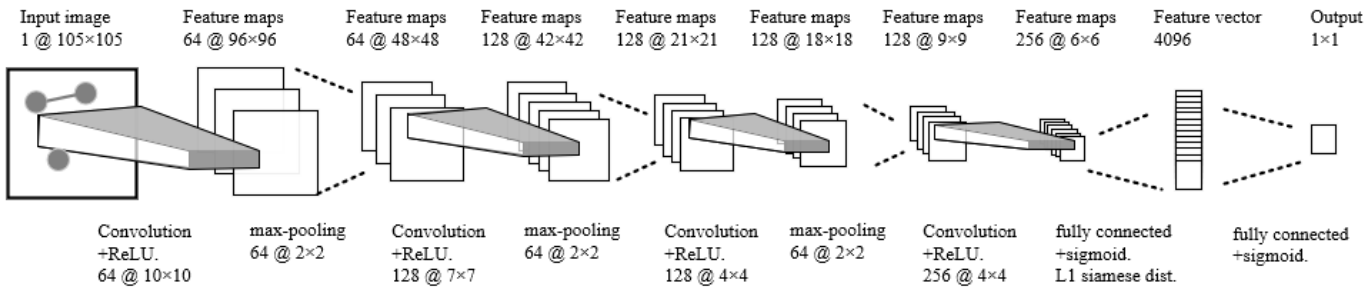


Fig. 3. Eight deep convolutional Siamese neural network using feature maps and max-pooling.

linear combination of class attentions on S; for x_i which is farthest from $\hat{X}$, its attentions under a certain measure is 0, and its value is the weight fusion of the corresponding labels corresponding to x_i which is similar to $\hat{X}$. The attentions mentioned above are embedding the training sample x_i and the test sample $\hat{X}$ separately; and then are inputed by software-max transforms, such as:

$$a\left(\hat{X}, x_i\right) = \frac{e^{c\left(f(\hat{x}), g(x_i)\right)}}{\sum_{j=1}^{k} e^{c\left(f(\hat{x}), g(x_j)\right)}} \tag{3}$$

where, c is the cosine distance. Two of the embedding models are shared, such as CNN. a is related to metric learning. For the sample x to be classified, it needs to be aligned with those labeled y, and other misalignments. Furthermore, the support set sample embedding model g can continue to be optimized, and the support set sample should be used to modify the embedding model f of the test sample. This can be solved in the following two aspects: i) embedding based on two-way LSTM learning training set, so that embedding of each training sample is the function of other training samples; ii) embedding of test samples based on attention-LSTM, so that embedding of each test sample is the function of embedding of training set. This work calls Full-Conditional Embedding (FCE). Although the above classification is done on the whole sample of support set, the embedding used for cosine distance calculation is independent of each other. Therefore, this work changes the embedding of support set samples to $g(x_i, S)$, which is useful when x_j is very close to x_i. This model uses Bidirectional LSTM (BiLSTM), forward LSTM and the backward LSTM are combined into a BiLSTM. By doing so, we can effectively use past features (through forward state) and future features (through backward state) within a specified time frame. We use back-propagation through time (BPTT) to train a bidirectional LSTM network. Over time, the forward and backward transfers on the expanded network are similar to the forward and backward transfers in conventional networks, except that we need to expand the hidden state for all time steps. We also need special processing at the beginning and end of the data points. Supposed that S is a random sequence, and then codes each x_i:

$$g(x_i, S) = \vec{h}_i + \overleftarrow{h}_i + g'(x_i) \tag{4}$$

$$(h_i, c_i) = LSTM\ (g'(x_i), \vec{h}_{i-1}, \vec{c}_{i-1}) \tag{5}$$

$$\left(\overleftarrow{h}_i, \overleftarrow{c}_i\right) = LSTM\ (g'(x_i), \overleftarrow{h}_{i-1}, \overleftarrow{c}_{i-1}) \tag{6}$$

where, $g'(x_i)$ is originally only dependent on its own embedding, x_i is for information exchange by BiLSTM. On the optimization of f, support set samples can be used to modify the embedding model of the test samples. This can be solved by a fixed-step LSTM and attention model of support set as:

$$f(\hat{x}, S) = atLSTM\ (f'(x), g(S), K) \tag{7}$$

$$\hat{h}_k, c_k = LSTM\ (f'(\hat{h}), [h_{k-1}, r_{k-1}], c_{k-1} \tag{8}$$

$$h_k = \hat{h}_k + f'(\hat{x}) \tag{9}$$

$$r_{k-1} = \sum_{i=1}^{|S|} a\left(h_{k-1}, g\ (x_i)\right) g\ (x_i) \tag{10}$$

$$a\left(h_{k-1}, g(x_i)\right) = soft\max(h_{k-1}^T\ g(x_i)) \tag{11}$$

where, $f'(x)$ is only dependent on the characteristics of the test sample itself; as the input of LSTM (unchanged at each step), K is the step of LSTM, $g(S)$ is the embedding of support set. As a result, the model ignores some samples in support set S. The embedding functions f and g optimize the feature space to improve the accuracy.

D. Insufficient Data-set Transfer Learning Modeling

For the convenience of introducing our zero-sample problem, here we first briefly introduce the recognition problem which is usually referred to in computer vision. We can summarize a simple pattern. First, we choose a suitable learning machine: convolutional neural network (CNN), prepare enough training data and corresponding labels for the learning machine, and use training data to train the learning machine. The main feature of the Convolutional neural network (CNN) is the use of convolutional layers. This is actually a simulation of the human visual neuron. A single neuron can only respond to certain specific image features, such as horizontal or vertical edges. It is very simple in itself, but these simple neurons form a layer. After the number of layers is enough, the system can obtain sufficient features. CNN is the most popular neural network model for image classification problems. An important idea behind CNN is the local understanding of the image. Its parameters will greatly reduce the time required for learning and reduce the amount of data required to train the model. The CNN has enough weights to see small patches of the image, rather than a fully connected network of weights from each pixel.

In practical use, the test data is input into the trained learner, and the learner can output the corresponding label of the test data. It is worth mentioning here that in this general pattern, the learner can only predict the existing categories in the training set when testing.

1. Basic Modeling

The concept of zero-shot learning and the initial solution, called attribute migration between classes, are given. We need to create a table to record the values of attributes of each category. We can map these pictures to the classification attribute table by looking up the table. These attribute descriptions can be summarized in advance because it is difficult to collect many pictures of unknown categories, but it is feasible to summarize the corresponding attributes of each category only. The core idea of attribute transfer between classes is that although objects have different categories, they have the same attributes, extract the attributes corresponding to each category and use several learners to learn. In the process of testing, the attributes of test data are predicted, and then the predicted attributes are combined with corresponding categories to realize the category prediction of test data. A simple model of zero sample learning can be summarized as shown in Fig. 4. The image space and label space are the initial image space and label space respectively. In zero-sample learning, images are usually mapped into feature space by some methods, which is called feature embedding. The same label is mapped into a label embedding, learning the linear or non-linear relationship between feature embedding and label embedding. Predictive transformation during testing replaces previous direct learning from images space to label space.

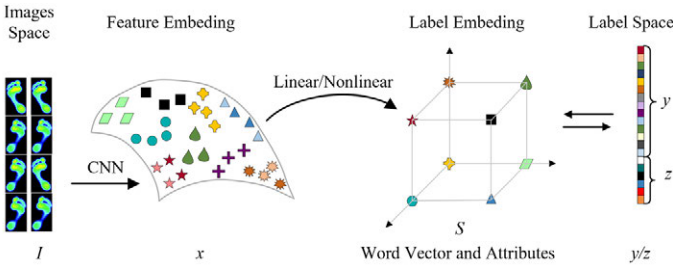


Fig. 4. Zero-shot learning using word vector and attributes.

2. Discriminative Learning of Latent Feature

Compared with the inter-class attribute migration method, it mainly adds two ideas: designs a special network to find the main part of the target in the picture; and designs the attributes using manual annotation, and adds label embed to search potential attributes from the original picture. The model diagram is shown in Fig. 5.

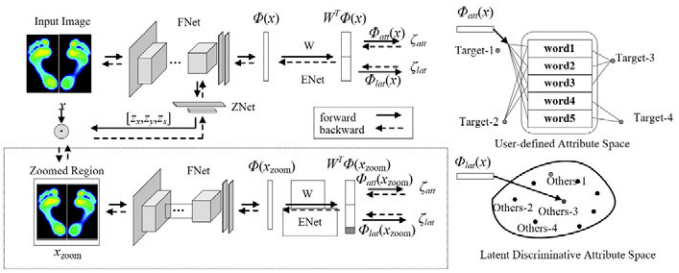


Fig. 5. Discriminative learning of latent feature-based modeling.

3. Improved Framework for the Proposed Model

Universal Zero Sample Learning (ZSL) requires accurate prediction of categories that have been seen (WS) and have not been seen (WU) (ZSL only requires prediction of categories that have not been seen). The main idea of this method is to train a WS first by using training data and training categories [37]–[39]. Then WS trains a graph convolution network using word vectors as input, and uses the graph convolution network to input the word vectors of test categories to get the WU. WS and WU together form a learner W to predict all categories, as shown in Fig. 6.

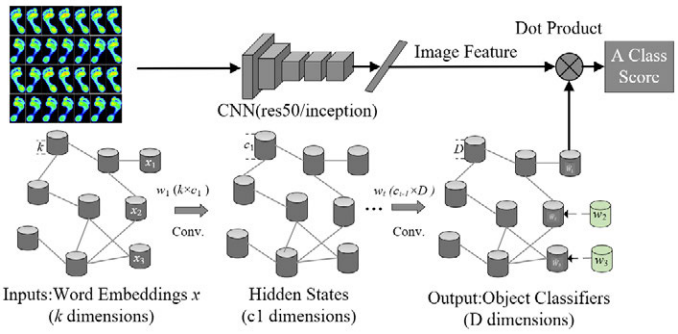


Fig. 6. Semantic (word embeddings) classifiers for plantar pressure image

III. Semantic Classification of Plantar Pressure Image Results

A. Semantic Classification of Plantar Pressure Image Results

Training a good CNN model for image classification requires not only computational resources but also a long time, especially when the model is complex and the amount of data is large. Standard desktop computers typically need to be trained for a few days if they are immobile. In order to train our plantar pressure image classifier quickly, we can use the model parameters that others have trained, and train our model on this basis. This belongs to transfer learning. Images from the basic data-set obtained are shown in Fig. 7.

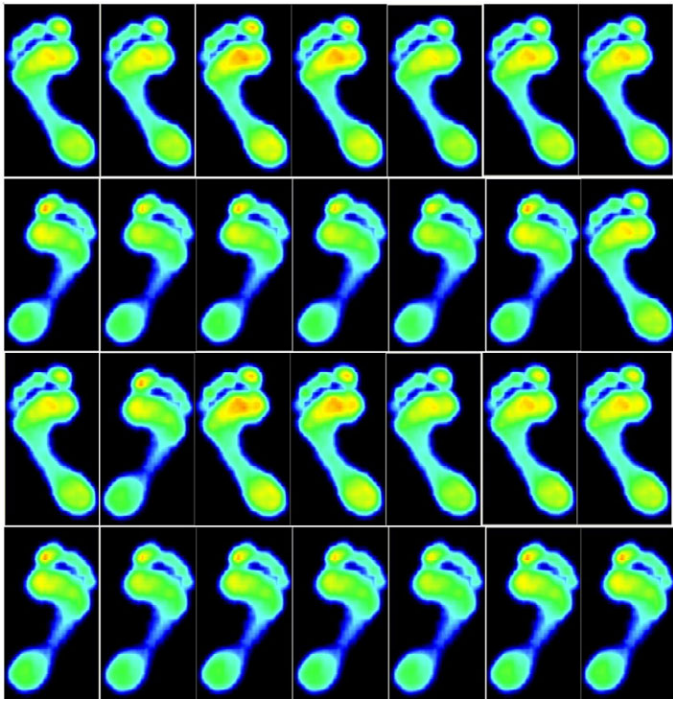


Fig. 7. Basic data-set acquired from the pressure sensors.

B. Pre-trained CNN using AlexNet for Classifiers

The data are divided into a training data set and a validation data set. 70% of the images were used for training, 15% for testing and 15% for verification. In the AlexNet system, SplitEachLabel divides image data stored into two new data storage areas, i.e. [imdsTrain, imdsValidation] = splitEachLabel (imds, 0.7,'randomized'); this small data set contains 55 training images and 20 validation images. AlexNet was trained on more than a million images, which can be grouped into 1,000 object categories. Therefore, the model has learned a rich feature representation based on many images. Let Net=alexnet and by using "analyzeNetwork", the network architecture and detailed information about the network layer can be presented visually and interactively. The steps are as follows:

Step 1: The first layer (image input layer) requires an input image of 277×277×3 in size, where 3 is the number of color channels.

Step 2: Replace the final layer. The last three layers of the pre-training network net are configured for 1000 classes. These three layers must be fine-tuned for the new classification problem. All layers except the last three layers are extracted from the pre-training network. By replacing the last three layers with a full connection layer, soft Max layer and classification output layer, the layer is migrated to the new classification task. -The options for a new full connection layer are specified based on the new data. To make learning faster in the new layer than in the migrating layer, the full connection layer Weight Learn Rate Factor (WLRF) and Bias Learn Rate Factor (BLRF) values are increased.

Step 3 Training the network. The network requires the size of the input image to be 277×277×3, but the images stored in image data have different sizes. The size of training image can be automatically adjusted by using enhanced image data storage. The training image in both horizontal and vertical directions is up to 30 pixels. Data enhancement helps to prevent network over-fitting and memorizes the details of training images. Specify training options. For migration learning, the shallow features of the pre-training network (migration layer weight) need to be retained. This combination of learning rate settings only accelerates the learning speed in the new layer, but slows

down the learning speed in other layers. When implementing transfer learning, the number of training rounds required is relatively small. A round of training is a complete training cycle for the whole training data set. Small batch sizes and validation data are specified. The software verifies the network every Validation Frequency iteration in the training process.

Step 4: Training the network consisting of migration layer and new layer.

Step 5: Calculate the classification accuracy for the verification sets.

Fig. 8 and Fig. 9 present the accuracy and loss of 25 layers of the CNN in AlexNet.

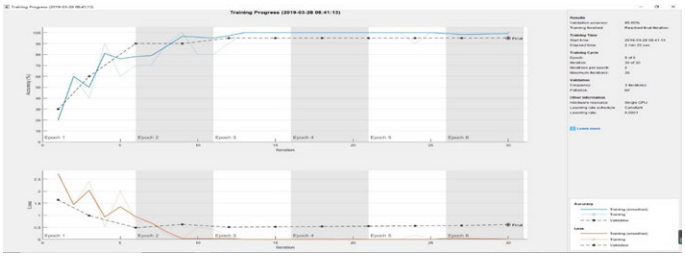


Fig. 8. Accuracy as a function of iteration in 25 layers of the CNN in AlexNete pressure sensors.

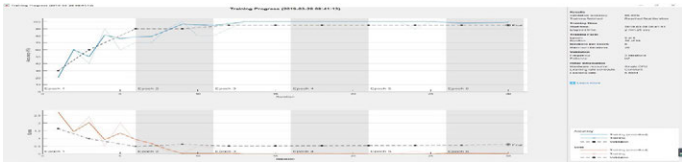


Fig. 9. Loss as a function of iteration in 25 layers of the CNN in AlexNet.

C. CNN based Transfer Modeling

The CNN model has two parts, the convolution layer in front and the full connection layer in the back. The function of the convolution layer is to extract image features while the function of the full connection layer is to classify features. Our method is to modify the full connection layer and retain the convolution layer on the inception-v3 network model. The parameters of convolution layer are trained by others. The parameters of the full connection layer require to initialize and use our own data to train and learn, as shown in Fig. 10.

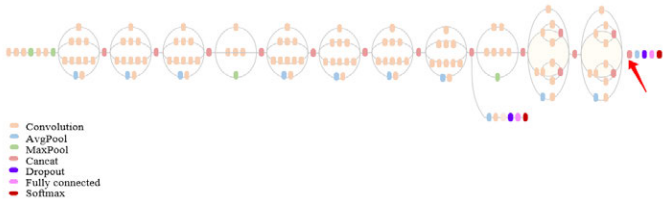


Fig. 10. Loss as a function of iteration in 25 layers of the CNN in AlexNet.

The front part of the red arrow in the inception-v3 model in Fig. 10 is the convolution layer, and the back part is the full connection layer. We need to modify the full connection layer and change the final output of the model to 5. Because the TensorFlow framework is used here, we need to get the tensor BOTTLENECK_TENSOR_NAME (output value of the last convolution activation function, number 2048) and the tensor JPEG_DATA_TENSOR_NAME of the initial input data of the model. The purpose of acquiring these two tensors is to obtain the image trained data through JPEG_DATA_TENSOR_NAME tensor input model and BOTTLENECK_TENSOR_NAME tensor after the convolution layer. By inception-v3 model including pre-trained parameters, the 2 tensors above mentioned are acquired. For the classification issue, the full connected layers need to be adjusted and here only one layer is added; the input data is BOTTLENECK_ TENSOR_NAME; finally, the cross-entropy loss function is defined. Because the training data set is relatively small, all pictures are input into the model through JPEG_DATA_TENSOR_NAME tensor, and then the values of BOTTLENECK_TENSOR_NAME tensor are obtained and saved to the hard disk. During model training, the value of the saved BOTTLENECK_TENSOR_NAME tensor is read from the hard disk as the input data of the full connection layer, the value of tensor of BOTTLENECK _TENSOR_NAME is acquired by the inputting images. The final accuracy of the test was found to be 93.5% over 4180 steps.

D. Discussion

As a result, the pre-trained CNN to assess the required level achieves performance improvements in plantar pressure image data-set, and the classification was also incrementally fine-tuned. Detection and characterization of plantar pressure image data-set by using deep learning with indices of Training-Testing Data Splitting (TTDS), Class Type (CT), Area Under the Curve (AUC), Average Precision Score (APS), recall, precision, f1 score (an index used to measure the accuracy of binary classification model in statistics -it considers both the accuracy and recall of the classification model), N, UN, Ave are compared and summarized in Table I [37],[40]-[42]. Table II shows the details of the public test database of SUN, CUB, AWA1, WAW2, and aPY [37]. Scaling and shifting (SS) is the known partitioning method while plain slot (PS) is the unknown partitioning method, tr is precision on known classification, ts is precision on unknown classification. The training and evaluation time for SS and PS are shown in Table III. The performance of the proposed method for public data-set of SUN CUB AWA1, AWA2, aPY [25] are shown in Table IV. By using inception-v3, the proposed model was also compared by using a public test database which H is a comprehensive index of "ts" and "tr", as shown in Table V. Therefore, the proposed CNNTM has better performance on primary databases on known classification.

IV. Conclusion

Large-scale methods are currently used to process most of the labeled data, but existing problems include lack of knowledge and many training samples. In practice, it is difficult to accumulate data for most categories and large-scale methods are not fully applicable. Therefore, learning many data-sets for a certain category is not feasible. While in each situation, a new category can be learned quickly with only a small number of samples. There are two main solutions under consideration at present: the first is that an object has not been seen before which is named as zero sample learning; and the second is to learn knowledge from existing tasks and apply it to future model training, which can be considered as a problem of Transfer Learning (TL). The method of the transfer-based learning refers to the use of these auxiliary data sets for transfer learning when there are other data sources. The idea is to learn a generalized representation from the data which can be directly used in the target data and small sample category learning process. Sample-based transfer learning is an attempt to give new weight to each sample in the source data so that it can better serve the new learning task. Samples that are more like the target data are selected from the source data to participate in the training, while samples that are not like the target data are excluded. The insufficient data-set learning methods are introduced, and an improved framework of transfer learning model is proposed and compared by other typical methods. The plantar pressure image data-set semantical classification issue performs high effectiveness with indices that were defined by current researchers.

TABLE I. Classifiers of VGG, ResNet, AlexNet and Pre-trained CNN for Plantar Pressure Image Data-set

Classifier	Training-Testing Data Splitting	Class Type	Precision	Recall	F1 Score	Accuracy	AUC	APS
		N	0.910	0.912	0.920			
	90-10%	UN	0.910	0.914	0.920	92.21	93.12	90.50
		Avg/Total	0.910	0.908	0.920			
		N	0.915	0.914	0.919			
VGG16+LR	80-20%	UN	0.915	0.918	0.918	91.12	92.10	90.72
		Avg/Total	0.915	0.917	0.918			
		N	0.915	0.912	0.907			
	70-30%	M	0.912	0.909	0.907	91.34	91.21	93.12
		Avg/Total	0.905	0.908	0.911			
		N	0.910	0.909	0.911			
	90-10%	UN	0.905	0.909	0.911	90.52	92.50	92.64
		Avg/Total	0.910	0.909	0.912			
		N	0.875	0.912	0.901			
VGG19+LR	80-20%	UN	0.872	0.909	0.902	91.28	91.34	91.12
		Avg/Total	0.875	0.904	0.912			
		N	0.870	0.880	0.899			
	70-30%	UN	0.912	0.889	0.892	92.10	86.12	90.45
		Avg/Total	0.910	0.878	0.878			
		N	0.835	0.812	0.832			
	90-10%	UN	0.890	0.831	0.823	80.32	86.22	91.25
		Avg/Total	0.850	0.835	0.825			
		N	0.835	0.848	0.845			
ResNet50+LR	80-20%	UN	0.869	0.879	0.889	80.15	83.24	86.25
		Avg/Total	0.889	0.877	0.887			
		N	0.825	0.808	0.832			
	70-30%	UN	0.824	0.811	0.821	80.25	81.25	86.75
		Avg/Total	0.829	0.812	0.821			
		N	0.908	0.870	0.899			
	90-10%	UN	0.908	0.914	0.883	92.32	93.07	92.65
		Avg/Total	0.909	0.921	0.883			
		N	0.912	0.919	0.876			
Pre CNN25+AlexNet	80-20%	UN	0.908	0.909	0.912	92.66	90.07	92.12
		Avg/Total	0.909	0.908	0.912			
		N	0.914	0.911	0.921			
	70-30%	UN	0.889	0.900	0.919	92.65	93.01	93.11
		Avg/Total	0.917	0.911	0.919			
		N	0.925	0.938	0.943			
	90-10%	UN	0.929	0.940	0.942	91.75	92.66	94.21
		Avg/Total	0.930	0.935	0.940			
		N	0.927	0.937	0.922			
inception-v3-CNN+TM (+)	80-20%	UN	0.921	0.931	0.921	92.75	91.25	94.36
		Avg/Total	0.919	0.923	0.921			
		N	0.915	0.912	0.922			
	70-30%	UN	0.912	0.913	0.910	93.50	94.22	95.55
		Avg/Total	0.907	0.907	0.910			

+ proposed method, Normal(N) and Unnormal (UN)

TABLE II. Number of Classes (Y(ts): test classes, Y(tr): training classes)

Data-set	Size	Detail	Attributes	Y	Y(tr)	Y(ts)
SUN	M	F	102	717	580+65	72
CUB	M	F	312	200	100+50	50
AWA1	M	C	85	50	27+13	10
AWA2	M	C	85	50	27+13	10
aPY	S	C	64	32	15+5	12

TABLE III. Number of Images, Training Time and Evaluation Time of TR and TS

		At Training Time				At Evaluation Time			
		SS		PS		SS		PS	
Data-set	Total	Y(tr)	Y(ts)	Y(tr)	Y(ts)	Y(tr)	Y(ts)	Y(tr)	Y(ts)
SUN [43]	14340	12900	0	10320	0	0	1440	2580	1440
CUB [44]	11788	8855	0	7057	0	0	2933	1764	2967
AWA1 [45]	30475	24295	0	19832	0	0	6180	4958	5685
AWA2 [45]	37322	30337	0	23527	0	0	6985	5882	7913
aPY [46]	15339	12695	0	5932	0	0	2644	1483	7924

TABLE IV. Classification Precision on Public Data-Set Comparing with Current Methods

	SUN		CUB		AWA1		AWA2		aPY	
Methods	SS	PS	SS	PS	SS	PS	SS	PS	SS	PS
DAP [45]	38.9	39.9	37.5	40.0	57.1	44.1	58.7	46.1	35.2	33.8
IAP [45]	17.4	19.4	27.1	24.0	48.1	35.9	46.9	35.9	22.4	36.6
CONSE [47]	44.2	38.8	36.7	34.3	63.6	45.6	67.9	44.5	25.9	36.6
CMT [48]	41.9	39.9	37.3	34.6	58.9	39.5	66.3	37.9	26.9	28.0
SSE [49]	54.5	51.5	43.7	43.9	68.8	60.1	67.5	61.0	31.1	34.0
LATEM [50]	56.9	55.3	49.4	49.3	74.8	55.1	68.7	55.8	34.5	35.2
ALE [50]	59.1	58.1	53.2	54.9	78.6	59.9	80.3	62.5	30.9	39.7
DEVISE [47]	57.5	56.5	53.2	52.0	72.9	54.2	68.6	59.7	35.4	39.8
SJE [51]	57.1	53.7	55.3	53.9	76.7	65.6	69.5	61.9	32.0	32.9
ESZSL [52]	57.3	54.5	55.1	53.9	74.7	58.2	75.6	58.6	34.4	38.3
SYNC [53]	59.1	56.3	54.1	55.6	72.2	54.0	71.2	46.6	39.7	23.9
SAE [54]	42.4	40.3	33.4	33.3	80.6	53.0	80.7	54.1	8.3	8.3
GFZSL [55]	62.9	60.6	53.0	49.3	80.5	68.3	79.3	63.8	51.3	38.4
CNNTM(+)	**63.9**	59.6	**55.12**	49.9	79.2	**70.7**	**80.5**	**67.2**	**52.4**	**39.9**

TABLE V. Comparing Analysis on Inception-v3-CNN+TM (+) Noted as CNNTM (+)

Methods	SUN			CUB			AWA1			AWA2			aPY		
-	ts	tr	H	ts	tr	H	ts	tr	H	ts	tr	H	ts	tr	H
DAP	4.2	25.1	7.2	1.7	67.9	3.3	0.0	88.7	0.0	0.0	84.7	0.0	4.8	78.3	9.0
IAP	1 .0	37.8	1.8	0.2	72.8	0.4	2.1	78.2	4.1	0.9	87.6	1.8	5.7	65.6	10.4
CONSE	6.8	39.9	11.6	1.6	72.2	3.1	0.4	88.6	0.7	0.5	90.6	1.0	0.0	91.2	0.0
CMT	8.1	21.8	11.8	7.2	49.8	12.6	0.9	87.6	1.8	0.5	90.0	1.0	1.4	85.2	2.8
CMT2	8.6	28.0	13.3	4.7	60.1	8.7	8.4	86.9	15.3	8.7	89.0	15.9	10.9	74.2	19.0
SSE	2.1	36.4	4.0	8.5	46.9	14.4	7.0	80.5	12.9	8.1	82.5	14.8	0.2	78.9	0.4
LATEM	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	0.1	73.0	0.2
ALE	11.8	33.1	26.3	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	4.6	73.7	8.7
DEVISE	16.9	27.4	20.9	23.8	53.0	32.8	13.4	68.7	22.4	17.1	74.7	27.8	4.9	76.9	9.2
SJE	14.7	30.5	19.8	23.5	59.2	33.6	11.3	74.6	19.6	8.0	73.9	14.4	3.7	55.7	6.9
ESZSL	11.0	27.9	15.8	12.6	63.8	21.0	6.6	75.6	12.1	5.9	77.8	11.0	2.4	70.1	4.6
SYNC	7.9	41.3	13.4	11.5	70.9	19.8	8.9	87.3	16.2	10.0	90.5	18.0	7.4	66.3	13.3
SAE	8.8	18.0	11.8	7.8	54.0	13.6	1.8	77.1	3.5	1.1	82.2	2.2	0.4	80.9	0.9
GFZSL	0.0	39.6	0.0	0.0	45.7	0.0	1.8	80.3	3.5	2.5	80.1	4.8	0.0	83.3	0.0
CNNTM(+)	21.7	42.9	32.1	25.5	58.7	66.1	12.5	81.2	30.1	12.8	82.1	27.3	14.1	83.2	20.1

Current TL algorithms mainly want to improve the performance of the same probability y in the feature space at different times. In the plantar pressure image data-set, the morphological changes of the anterolateral prefrontal cortex and the medial prefrontal cortex, which belong to the visceral motor network system are found to be effective by plantar pressure structural imaging studies. The improved insufficient data-set learning method proposed in this work can solve the problem of insufficient training and accomplish excellent automatic recognition. While TL is mainly applied to small and volatile data sets (such as sensor data, text categorization, image categorization, etc.) in future research it is needed to consider how to use the TL method in a wider range of data scenarios.

Acknowledgment

This work is supported by Science Foundation of Ministry of Education of China (Grant No:18YJC760099) and Zhejiang Provincial Key Laboratory of Integration of Healthy Smart Kitchen System (Grant No: 19080049-N).

References

[1] J. Deng, et al, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848

[2] S. Pachori, A. Deshpande and S. Raman, "Hashing in the zero shot framework with domain adaptation," *Neurocomputing*, 275, 2137–2149. doi:10.1016/j.neucom. 2017.10.061

[3] Y. He, Y. Tian, and D. Liu, "Multi-view transfer learning with privileged learning framework," *Neurocomputing*, 335, 131–142. doi:10.1016/j.neucom.2019.01.019

[4] A. Lumini and L. Nanni, "Deep learning and transfer learning features for plankton classification," *Ecological Informatics*, 51, 33–43. doi:10.1016/j.ecoinf.2019.02.007

[5] A. Tavanaei, at al, "Deep learning in spiking neural networks," *Neural Networks*, 111, 47–63. doi:10.1016/j.neunet.2018.12.002

[6] H. Zhang, Y. Long and L. Shao, "Zero-shot hashing with orthogonal projection for image retrieval," *Pattern Recognition Letters*, 117, 201–209. doi:10.1016/j.patrec.2018.04.011

[7] A. M. Anam and M. A. Rushdi, "Classification of scaled texture patterns with transfer learning," *Expert Systems with Applications*, 120, 448–460. doi:10.1016/j.eswa.2018.11.033

[8] Y. Wu, Y. Su, and Y. Demiris, "A morphable template framework for robot learning by demonstration: Integrating one-shot and incremental learning approaches," *Robotics and Autonomous Systems*, 62, 1517–1530. doi:10.1016/j.robot.2014.05.010

[9] S. Cisse, *The Fortuitous Teacher*, pp. 119-143, 2016, Chandos Publishing.

[10] Y. Ji, Y. Yang, X. Xu, and H. T. Shen, "One-shot learning based pattern transition map for action early recognition," *Signal Processing*, 143, 364–370. doi:10.1016/j.sigpro.2017.06.001

[11] U. Mahbub, et al. "A template matching approach of one-shot-learning gesture recognition," *Pattern Recognition Letters*, 34, 1780–1788. doi:10.1016/j.patrec.2012.09.014

[12] X. Xu, S. Gong, and T. M. Hospedales, "Zero-shot crowd behavior recognition," pp. 341–369. In V. Murino, M. Cristani, S. Shah and S. Savarese (eds) *Group and Crowd Behavior for Computer Vision*. doi:10.1016/B978-0-12-809276-7.00018-7

[13] Z. Abderrahmane, G. Ganesh, A. Crosnier and A, Cherubini, "Haptic zero-shot learning: recognition of objects never touched before," *Robotics and Autonomous Systems*, 105, 11–25. doi:10.1016/j.robot.2018.03.002

[14] Z. Ji, et al., "Zero-shot learning with multi-battery factor analysis," *Signal Processing*, 138, 265–272. doi:10.1016/j.sigpro.2017.03.023

[15] Z. Ji, et al., "Manifold regularized cross-modal embedding for zero-shot learning," *Information Sciences*, 378, 48–58. doi:10.1016/j.ins.2016.10.025

[16] M. Liu, D. Zhang and S. Chen, "Attribute relation learning for zero-shot classification," *Neurocomputing*, 139, 34–46. doi:10.1016/j.neucom.2013.09.056

[17] B. Liu, et al., "Combining ontology and reinforcement learning for zero-shot classification," *Knowledge-Based Systems*, 144, 42–50. doi:10.1016/j.knosys.2017.12.022

[18] T. Long, et al., "Zero-shot learning via discriminative representation extraction," *Pattern Recognition Letters*, 109, 27–34. doi:10.1016/j.patrec.2017.09.030

[19] Y. Yu, Z. Ji, J. Guo and Y. Pang, "Zero-shot learning with regularized cross-modality ranking", *Neurocomputing*, 259, 14–20. doi:10.1016/j.neucom.2016.06.085

[20] A. Bear and D. G. Rand, "Modeling intuition's origins", *J. Applied Research in Memory and Cognition*, 5, 341–344. doi:10.1016/j.jarmac.2016.06.003

[21] S. Blaes and T. Burwick, "Few-shot learning in deep networks through global prototyping," *Neural Networks*, 94, 159–172. doi:10.1016/j.neunet.2017.07.001

[22] A. Chen, et al., "On the systematic method to enhance the epiphany Ability of individuals," *Procedia Computer Science*, 31, 740–746. doi:10.1016/j.procs.2014.05.322

[23] K. Dombroski, "Learning to be affected: Maternal connection, intuition and elimination communication," *Emotion, Space and Society*, 26, 72–79. doi:10.1016/j.emospa.2017.09.004

[24] J. Li, et al., "What is in a name?: The development of cross-cultural differences in referential intuitions," *Cognition*, 171, 108–111. doi:10.1016/j.cognition.2017.10.022

[25] X. Li, et al., "Zero-shot classification by transferring knowledge and preserving data structure," *Neurocomputing*, 238, 76–83. doi:10.1016/j.neucom.2017.01.038

[26] A. Soudani, and W. Barhoumi, "An image-based segmentation recommender using crowdsourcing and transfer learning for skin lesion extraction," *Expert Systems with Applications*, 118, 400–410. doi:10.1016/j.eswa.2018.10.029

[27] K. Deschamps, et al., "Efficacy measures associated to a plantar pressure-based classification system in diabetic foot medicine," *Gait and Posture*, 49, 168–175. doi:10.1016/j.gaitpost.2016.07.009

[28] J. A. Ramirez-Bautista, et al., "Review on plantar data analysis for disease diagnosis," *Biocybernetics and Biomedical Engineering*, 38, 2, 342–361. doi:10.1016/j.bbe.2018.02.004

[29] J. Sommerset, et al., "Plantar acceleration time: A novel technique to evaluate arterial flow to the foot," *Annals of Vascular Surgery*, 60, 308–314. doi: 10.1016/j.avsg.2019.03.002

[30] Y. Xia, et al., "A convolutional neural network cascade for plantar pressure images registration," *Gait and Posture*, 68, 403–408. doi:10.1016/j.gaitpost.2018.12.021

[31] C. Wang and G. Tzanetakis, "Singing style investigation by residual Siamese convolutional neural networks," in *Proc. ICASSP*, 2018, pp 116–120. doi:10.1109/ICASSP.2018.8461660

[32] J. Lu, et al., "Boosting few-shot image recognition via domain alignment prototypical networks," in Proc. ICTAI, 2018, pp. 260–264. doi: 10.1109/ICTAI.2018.00048

[33] J. Shi, et al., "Concept learning through deep reinforcement learning with memory-augmented neural networks," *Neural Networks*, 110, 47–54. doi:10.1016/j.neunet.2018.10.018

[34] T. L. Griffiths, et al., "Doing more with less: Meta-reasoning and meta-learning in humans and machines," *Current Opinion in Behavioral Sciences*, 29, 24–30. doi:10.1016/j.cobeha.2019.01.005

[35] S. Pramanik and A. Hussain, "Text normalization using memory augmented neural networks," *Speech Communication*, 109, 15–23. doi:10.1016/j.specom.2019.02.003

[36] A. E. Gutierrez-Rodríguez, et al., "Selecting meta-heuristics for solving vehicle routing problems with time windows via meta-learning," *Expert Systems with Applications*, 118, 470–481. doi:10.1016/j.eswa.2018.10.036

[37] Y, Xian, et al., "Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly," *IEEE Trans. Pattern Analysis and Machine Intelligence*, early-access. doi: 10.1109/TPAMI.2018.2857768

[38] S. J, Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Eng.*, 22, 1345–1359. doi: 10.1109/TKDE.2009.191

[39] B. Gong, Y. Shi, F. Sha and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in Proc. CVPR, 2012, pp. 2066–2073. doi:10.1109/CVPR.2012.6247911

[40] Shallu and R. Mehra, "Breast cancer histology images classification: Training from scratch or transfer learning?" *ICT Express*, 4, 247–254. doi:10.1016/j.icte.2018.10.007

[41] H. Hermessi, O. Mourali and E. Zagrouba, "Deep feature learning for soft tissue sarcoma classification in MR images via transfer learning," *Expert Systems with Applications*, 120, 116–127. doi:10.1016/j.eswa.2018.11.025

[42] P. Herent, et al., "Detection and characterization of MRI breast lesions using deep learning," *Diagnostic and Interventional Imaging*, 100, 219–225. doi:10.1016/j.diii.2019.02.008

[43] G. Patterson and J. Hays, "SUN attribute database: Discovering, annotating, and recognizing scene attributes," in *Proc. CVPR*, 2012, pp. 2751–2758. doi:10.1109/CVPR.2012.6247998

[44] P. Welinder, et al., Caltech-UCSD Birds 200. Computation & Neural Systems Technical Report CNS-TR: 1–15. https://authors.library.caltech.edu/27468

[45] C. H. Lampert, H Nickisch and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 36, 453–465. doi:10.1109/TPAMI.2013.140

[46] A. Farhadi, I. Endres, D. Hoiem and D. Forsyth, "Describing objects by their attributes," in *Proc. CVPR*, 2009, pp. 1778–1785. doi:10.1109/CVPR.2009.5206772

[47] N. M. Norouzi, et al., "Zero-shot learning by convex combination of semantic embeddings," arXiv:1802.10408.

[48] R. Socher, M. Ganjoo, C. D.Manning and A. Ng, "Zero-shot learning through cross-modal transfer," in *Proc. Int. Conf. Neural Information Processing Systems*, p.p. 935–943.

[49] Z. Zhang and V. Saligrama, "Zero-shot learning via semantic similarity embedding," in *Proc. ICCV*, 2015, pp. 4166–4174. doi: 10.1109/ICCV.2015.474

[50] Y. Xian, et al., "Latent embeddings for zero-shot classification," in *Proc. CVPR*, 2016, pp. 69–77. doi:10.1109/CVPR.2016.15

[51] Z. Akata, et al., "Evaluation of output embeddings for fine-grained image classification," in *Proc. CVPR*, 2015, pp. 2927–2936. doi:10.1109/CVPR.2015.7298911

[52] B. Romera-Paredes and P. H. S. Torr, "An embarrassingly simple approach to zero-shot learning," pp. 11–30. In: R. Feris, C. Lampert and D. Parikh (eds), *Visual Attributes. Advances in Computer Vision and Pattern Recognition*. doi:10.1007/978-3-319-50077-5_2

[53] S. Changpinyo, W. Chao, B. Gong and F. Sha, "Synthesized classifiers for zero-shot learning," in *Proc. CVPR*, 2016, pp. 5327–5336. doi:10.1109/CVPR.2016.575

[54] E. Kodirov, T. Xiang and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proc. CVPR*, 2017, pp. 4447–4456. doi:10.1109/CVPR.2017.473

[55] V. K. Verma and P. Rai, "A simple exponential family framework for zero-shot learning," in: M. Ceci, J. Hollmén, L. Todorovski, C. Vens and S. Džeroski (eds), *Machine Learning and Knowledge Discovery in Databases. Lecture Notes in Computer Science*, 10535. doi:10.1007/978-3-319-71246-8_48

Yao Wu

Yao Wu is an Assistant Professor of Department of Industrial Design, Wenzhou Business College, China. He holds a master's degree in software engineering from Huazhong University of Science and Technology in Wuhan, China, in 2014. and a Bachelor degree of consumer product design from Shaanxi University of Science and Technology, China, in 2005. His research interests include integrated innovation design and sustainable design.

Qun Wu

Qun Wu, is an Associate Professor of Human Factor at the Institute of Universal Design, Zhejiang Sci-Tech University, China. He received his Ph.D. in College of Computer Science and Technology from Zhejiang University, China, in 2008. He holds a B.E. degree in Industrial Design from Nanchang University, China, in 2001, and a M.E. degree in Mechanical Engineering from Shaanxi University of Science and Technology, China, in 2004. His research interests include machine learning, human factor and product innovation design.

Nilanjan Dey

Nilanjan Dey, was born in Kolkata, India, in 1984. He received his B.Tech. degree in Information Technology from West Bengal University of Technology in 2005, M. Tech. in Information Technology in 2011 from the same University and Ph.D. in digital image processing in 2015 from Jadavpur University, India. In 2011, he was appointed as an Asst. Professor in the Department of Information Technology at JIS College of Engineering, Kalyani, India followed by Bengal College of Engineering College, Durgapur, India in 2014. He is now employed as an Asst. Professor in Department of Information Technology, Techno India College of Technology, India. His research topic is signal processing, machine learning, and information security. Dr. Dey is an Associate Editor of IEEE Access and is currently the Editor-in-Chief of the International Journal of Ambient Computing and Intelligence, and Series co-editor of Springer Tracts of Nature-Inspired Computing (STNIC).

R. Simon Sherratt

R. Simon Sherratt (M'97–SM'02–F'12), received the B.Eng. degree in Electronic Systems and Control Engineering from Sheffield City Polytechnic, UK in 1992, M.Sc. in Data Telecommunications in 1994 and Ph.D. in video signal processing in 1996 from the University of Salford, UK. In 1996, he has appointed as a Lecturer in Electronic Engineering at the University of Reading where he is now Professor of Biosensors. His research topic is signal processing and communications in consumer devices focusing on wearable devices and healthcare. He is a reviewer for the IEEE Sensors Journal is now the Emeritus Editor-in-Chief of the IEEE Transactions on Consumer Electronics.

An Intelligent Technique for Grape Fanleaf Virus Detection

Mojtaba Mohammadpoor[1]*, Mohadese Gerami Nooghabi[2], Zahra Ahmedi[1]

[1] Electrical & Computer Engineering Department, University of Gonabad, Gonabad (Iran)
[2] Department of Agricultural and Natural Resources, University of Gonabad, Gonabad (Iran)

Received 8 April 2019 | Accepted 3 January 2020 | Published 3 February 2020

Abstract

Grapevine Fanleaf Virus (GFLV) is one of the most important viral diseases of grapes, which can damage up to 85% of the crop, if not treated at the right time. The aim of this study is to identify infected leaves with GFLV using artificial intelligent methods using an accessible database. To do this, some pictures are taken from infected and healthy leaves of grapes and labeled by technical specialists using conventional laboratory methods. In order to provide an intelligent method for distinguishing infected leaves from healthy ones, the area of unhealthy parts of each leaf is highlighted using Fuzzy C-mean Algorithm (FCM), and then the percentages of the first two segments area are fed to a Support Vector Machines (SVM). To increase the diagnostic reliability of the system, K-fold cross validation method with k = 3 and k =5 is applied. After applying the proposed method over all images using K-fold validation technique, average confusion matrix is extracted to show the True Positive, True Negative, False Positive and False Negative percentages of classification. The results show that specificity, as the ability of the algorithm to really detect healthy images, is 100%, and sensitivity, as the ability of the algorithm to correctly detect infected images is around 97.3%. The average accuracy of the system is around 98.6%. The results imply the ability of the proposed method compared to previous methods.

Keywords

Artificial Neural Networks, SVM, Fuzzy C-mean Algorithm, Grape Fanleaf Virus Detection, Grape Diseases.

DOI: 10.9781/ijimai.2020.02.001

I. Introduction

Recent advances in agricultural technology have led to increase demand for development of non-destructive diagnostic methods. Spectrometry and imaging techniques are approaches used to detect disease and stress in trees and plants. By using image processing and machine learning methods can accurately detect and diagnose the disease at very low cost, and increase production.

Vine diseases are diverse, but the Grape Fanleaf Virus (GFLV) is the most harmful grape loss in the world, with losses of up to 85% of the crop. The disease has been reported in most temperate regions of the world. There are several common laboratory methods for identifying viruses, for instance Kaur et al. have conducted a study of various computer vision applications that classify images of plant leaves to detect diseases [1]. Lots of researchs on the use of neural networks in detection of plant diseases has been performed. For example, Beeshish et al. in [2] have classified hereditary diseases using a post-propagation neural network. Mahmoudi et al. have used visual machine techniques to evaluate the color and appearance properties of the leaves and used these properties to identify two walnut diseases. They have reached to 95% accuracy in classifying the plant diseases [3]. Due to the small differences between the infected and healthy images, the convolutional neural network (CNN) has used to identify plant diseases in recent years [4]-[5]. Due to the complexity of these networks and their needs for many images for training, common neural networks are still intresting. For example, Shah, Nikhil and Sarika Jain in [6] have used artificial neural networks to detect cotton leaf diseases. In [7] Hosseini et al. have provided a system to detect fungal infection of white fish powder and anthracnose of cucumber leaves with image processing techniques and artificial neural networks. Their method was consisted of three steps: segmentation, separation of damaged parts of the leaf and classification.

In [8] Omrani et al. have proposed an Adaptive Neuro-Fuzzy Inference System (ANFIS) to predict four varieties of apple plants by processing their leaf images. For this purpose, after collecting image dataset of leaf samples, they extracted morphological, color and texture features. Their results showed that ANFIS could classify the leaves successfully. The precision of their method for experimental classification was between 83% to 95%. Al-Hiary et al. have classified the leaf symptoms of the diseases by using K-means clustering and neural networks [9]. Menukaewjinda et al. in [10] has used the backward propagation neural networks (BPNN) for a competent grape leaf color, but not for a specific grape disease. Subsequently, several studies were conducted to develop this method and algorithms, but all were based on specific symptoms of plant diseases and not a defined disease, such as viruses [11]. Dubey, et al. in [12] have proposed K-Means clustering segmentation technique to detect infected fruit part. Belkhodja and Hamdadou have also proposed a computer aided detection system for detecting breast masses in [13]. Pujari, Yakkundimath and Byadgi have used SVM and ANN for classification of plant disease in [14].

In this paper, we first discuss the virus, its identification and its experimental diagnostic methods in sections II. By introducing useful

* Corresponding author.
E-mail address: mohammadpur@gonabad.ac.ir

techniques, including segmentation, classification and cross-validation in section III, the proposed method will be described based on them in section IV. The results will be displayed and discussed in section V, and the document will be concluded in section VI.

II. Disease Description

The Grapevine fanleaf virus (GFLV) is one of the most important grape diseases that causes the leaves to be severely distorted, asymmetrical, hollowed out and wrinkled, and exhibit sharp dentures. Other symptoms of this virus are yellowish color and delayed veins [15]-[17]. The virus is transmitted by an ectoparasite nematode - (Xiphinema Index) [18]. The virus is widespread around the world but its origin is Iran [19]-[20].

There are some Virus detection methods. Some of the common methods used in most of laboratories are discussed in this section.

A. Electron Microscope

Electron microscope method was used to observe virions and detect them in the vector [20]. Due to the low concentration and heterogeneous distribution of the virus in plant tissues, this method has not been used for identification [15].

B. Use of Indicator Plants

One of the methods used to detect viral diseases of plants are indicator plants [21]. The best indicator for detecting GFLV is Vitis rupestris St. George, which is a mosaic marker, and can determine oily spots, deformities and felt leaves [22]. However, this method takes a long time and does not cause symptoms at low virus concentrations.

C. Serological Methods

The concentration of GFLV in the warm season is significantly reduced in plant tissues, hence viral concentrations are below the detection threshold of serological tests [20].

D. Enzyme-linked Immunosorbent Assay

Enzyme-linked immunosorbent assay (ELISA) is one of the most commonly used methods for detecting GFLV in plant tissues [23]. In this technique, the virus in the plant can be determined based on the antigenic tendency of an antigen to attach to a specific antibody [24]. Due to higher virus concentrations in young tissues, the use of young leaves for this method is better than in older ones [20]. Despite the effectiveness of this method, it is not practically applicable in many cases for the diagnosis of plant viruses due to its time consuming procedure [25].

E. Molecular Methods

Reverse Transcriptase- Polymerase Chain Reaction (RT-PCR) is the most sensitive method for detecting GFLV. Unlike other methods of identification, which depend on the age of the tissue, variety, season and virus concentration, PCR have the ability to accurately diagnose the disease independently.

Phenolic and polysaccharide compounds in grape leaves limit the release of pure RNA virus, therefore, the ability of PCR will be reduced. Despite significantly improvements in genomic extraction methods, it cannot be said that their errors reached to zero [20]. Hybridization of a nucleic acid using a probe is another sensitive method for detecting GFLV [22]. To achieve this, special probes against the genome of the virus have been developed [23]. Also this method has some disadvantages. Considering the fact that the intensity of reaction of the probes is low due to the concentration of the virus, tracking would be difficult in the warm seasons, when the concentration of the virus in the tissues decreases.

III. Materials and Methods

A digital image is actually a two-dimensional signal or, in other words, a data matrix that is created by measuring the reflected light from an object. Each of the image components is called a pixel. In gray level images, the minimum pixel value is zero, which represents a completely black dot. The maximum of each pixel is usually 255, which represents a completely white pixel. In color images using the RGB color standard, each pixel has three values ranging from 0 to 255, indicating the red, green, and blue colors, respectively.

A. Clustering

Clustering is an unsupervised learning method, which plays an important role in data mining, machine learning, and pattern recognition. This is an information process that divides data samples into several categories, called clusters, based on similarities between them. Various criteria can be considered as a measure of similarity, for example, one can use the distance criterion for clustering and consider objects closer to each other as a cluster. This type of clustering is called distance clustering. Clustering methods can be divided into hierarchical and separated delimiters. Hierarchical algorithms use the similarity criterion, and at each stage they divide the data into two categories and ultimately create a tree structure of this unit called dendrogram. Separation algorithms directly group data into several clusters. These algorithms are divided into hard (or exclusive) and soft (or fuzzy) algorithms. In strict clustering, the input sample belongs only to a cluster, while in soft clustering, its membership for each cluster is determined by a number between zero to one [26].

B. Fuzzy Clustering and FCM Algorithm

The FCM Clustering Algorithm is the basic algorithm of the segmentation methods. This algorithm is always of interest to researchers because of its advantages, such as its simple structure, ease of implementation, fast convergence and its need for a small storage space. Its simplicity is due to the fact that each cluster is represented by a center of gravity or an average value. One of the drawbacks of this algorithm is that the weight of the functions is constant through the entire clustering process. To overcome this drawback, various strategies have been proposed for adapting the weight of functions during the clustering process. In [27] Zhi et al. have developed a clustering algorithm based on C-means with automatic weighting functions during clustering.

C. Support Vector Machines (SVM)

The most common neural network techniques often focus on improving the structure of the neural network, in order to minimize estimation error and the number of neural network errors, but in a specific form of them, known as Support Vector Machines (SVM), it focuses solely on reducing the operational risk associated with inadequate performance. The SVM network structure has much in common with Multilayer Perceptron (MLP) neural networks, and in practice the main difference is in the learning style. Since this method is used in this paper, it is discussed in more detail below [29]. The support vector machines in their simplest form, linear SVM, consist of a cloud page that separates positive and negative sample sets with maximum distances (Fig. 1).

In general, this problem can be considered in an n-dimensional space in which the data are divided into two categories. In this case, instead of separating lines, a hyperplane separator will be used. In general, in the hyperplane type separator we will have:

$$\sum w_i x_i + b = 0 \tag{1}$$

It can be expressed in the following way:

$$u = \vec{w} \,.\, \vec{x} + b \tag{2}$$

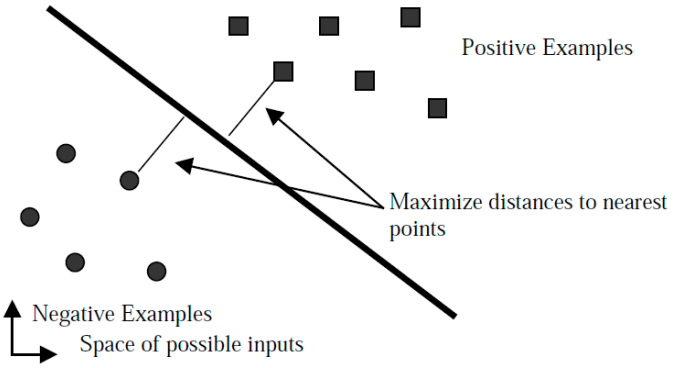


Fig. 1. SVM as a cloud page for linear separation of samples in data space [29].

Where w is the vector weight of perpendicular vector to the hyperplane and b is the initial value. In this display, $u = 0$ refers to the separator line itself, and the closest points are located on the plates $u = \pm 1$. In fact, with the assumption of the separation of two positive and negative data classes, the boundary vectors will be placed on the following hyperplanes:

$$\vec{w} \cdot \vec{x} + b = \pm 1 \tag{3}$$

The area between these two hyperplanes is called the margin. Fig. 2 shows the two-dimensional state with the assumption that the initial value is negative.

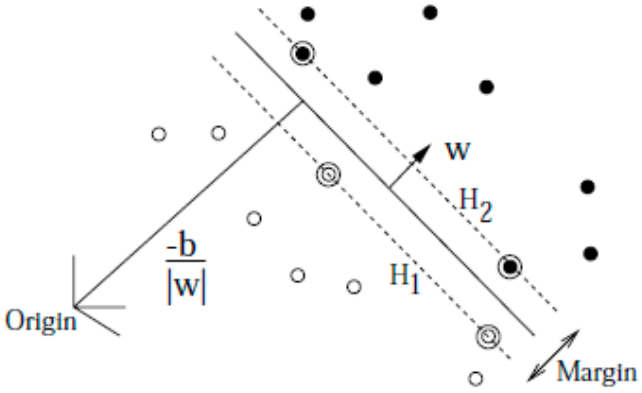


Fig. 2. Two-dimensional distance [25].

As shown in Fig. 2, space is divided into two categories of samples with the following characteristics:

$$\vec{w} \cdot \vec{x} + b \geq +1 \text{ for } y_i = 1 \tag{4}$$

$$\vec{w} \cdot \vec{x} + b \leq -1 \text{ for } y_i = -1 \tag{5}$$

The above formula can be combined as follows:

$$y_i (\vec{w} \cdot \vec{x} + b) - 1 \geq 0 \ \forall i \tag{6}$$

In this case, the distance to the source vertically (the closest distance) for the points on the hyperplane $\vec{w} \cdot \vec{x} + b = 1$ is:

$$|1 - b| / |w| \tag{7}$$

And in the same way, the distance from the source vertically to the points on the cloud plate $\vec{w} \cdot \vec{x} + b = -1$ is equal to:

$$|-1 - b| / |w| \tag{8}$$

On the other hand, the source distance to the separator hyperplane is equal to:

$$|b| / |w| \tag{9}$$

So the smallest distance between this hyperplane and any of the pages is as follows:

$$d_+ = d_- = 1 / |w| \tag{10}$$

Therefore, the margin, as the distance between the two superimposed pages is as follows:

$$d_+ + d_- = 2 / |w| \tag{11}$$

In this way, the maximization of the margin can be expressed in the form of the following optimization equation:

$$\min |w|^2$$

$$\text{subject to: } y_i (\vec{w} \cdot \vec{x} + b) - 1 \geq 0 \ \forall i \tag{12}$$

D. K-fold Validation Method

There are various methods for validating the algorithms applied on databases to ensure that the results are valid in all circumstances and are not depend on selection of training and testing parts. K-fold is one of the common methods for validating classification algorithms. In this method, training and testing data are divided into K subsets. Of the subsest, one is left for validation, and K-1 substes are used for training. This procedure is repeated K times so that all data is used exactly once for validation purposes. Finally, the final result is considered as the average of all round results [30].

IV. Proposed Method

In order to detect GLFV disease using image processing, a method is proposed and implemented in accordance with the flowchart shown in Fig. 3. Accordingly, after collecting practical images, some preliminary processing steps are performed on them, for example their background is removed, and their image intensity is improved. Images are divided into two categories: healthy and infected leaves, based on visual inspection by experts, and confirmed by molecular testing by RT-PCR. Then, the FCM algorithm is performed to segment the images. As the separation of healthy parts from unhealthy on a green page is better, this color page is considered as the input to the next step of the proposed algorithm.

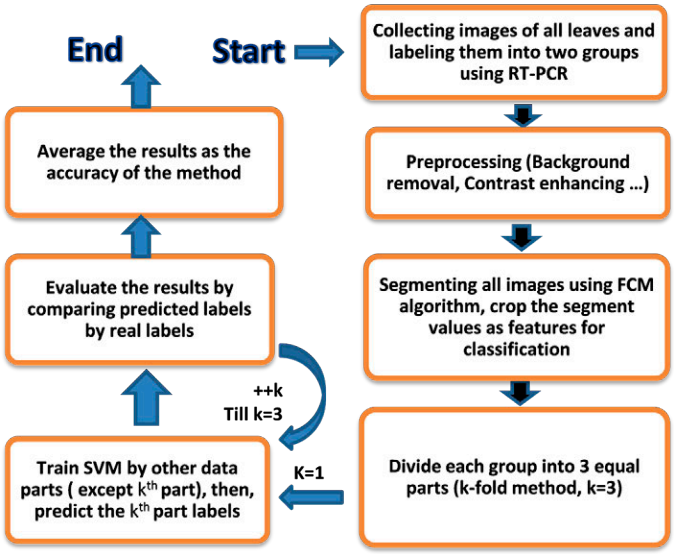


Fig. 3. Flowchart of the proposed method.

A. Data Collection and Validation

The leaves used in this research belong to the Kashmar region of Khorasan-Razavi province, Iran, and were captured by a Sony DSC-N2 camera during Spring and May 2013. Leaves are divided into two groups (healthy and infected) based on the results of molecular testing with RT-PCR at at Ferdousi University of Mashhad laboratory. Totally 92 images are collected of healthy and infected leaves. Examples of these images are shown in Fig. 4.

Fig. 4. Right leaf: healthy, Left leaf: completely infected with the virus.

V. Results and Findings

The result of applying different parts of the proposed algorithm is presented in this section.

A. Background Removal

Due to the small changing of criteria such as light, background color and camera height, etc. during taking pictures of leaves, the background of the image may cause some difficulties, hence all leaves backgrounds are deleted. Fig. 5 shows a sample image before and after of background removing procedure.

(a)

(b)

Fig. 5. (a) a sample of original image of a partially infected leaf, (b) the same image after removing its background.

B. Contrast Improving of Images

Improving the intensity of an image is usually desirable. Its use in this research is to improve the characteristics of yellow and cream mosaics, as clear signs of a viral disease in infected leaves. A sample image after improving the intensity is shown in Fig. 6. Comparing Fig. 6 and Fig. 5, one can clearly see the difference between the two images. In Fig. 6 yellow and cream spots in the image are brighter than other parts.

Fig. 6. Sample infected leaf image after contrast improving.

C. Applying Segmentation Algorithms

In this section, the FCM algorithm is applied to the green plane of the image of previous stage. The number of sections is considered as three. Fig. 7 shows the results of the algorithm.

(a)

(b)

Fig. 7. The result of applying FCM segmentation algorithm to an infected image, a) Segment I, b) Segment II.

As shown in Fig. 7 and in comparison with Fig. 6, the contaminated parts of image appeared in varying degrees in the first and second parts, so that the completely contaminated parts are appear in the first segment, and the semi-infected parts in the second segment. Viewing the actions of the algorithm on some other healthy or infected images indicates the ability of FCM algorithm to distinguish infected parts from healthy parts. Therefore, it seems that the percentage of infected and healthy parts in different segments can be considered as a decision factor for classifying the leaves into healthy or infected categories. Fig. 8 shows the percentage of the two segments for all images. As implied, all infected leaves (numbers 1-74) have some values in both segments. The average values are 31.5% and 60.5% for segments I, II, respectively. On the other hand, helthy leaves have some values in segment I, averagely 30.5%, while they don't have any value in segment II.

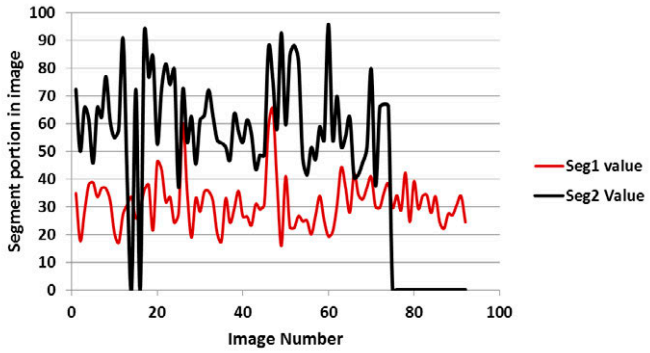


Fig. 8. Segments percentages of all leaves images.

D. Classification Results

In this section, the support vector machines algorithm (SVM) is performed over the previous stage dataset. To increase the reliability of the proposed algorithm, k-fold cross validation method with k = 3 is used. Based on this technique, the data in both classes are divided into 3 equal parts, and each time the two parts are used to train the network, and the remaining part is used for testing the model. This operation is repeated three times, so that all parts are considered for evaluation once. As a result, the mean value of the round results is calculated. The final results are presented in Table I.

TABLE I. Classification Results of Healthy and Infected Leaves (k = 3)

		predicted classes	
		Infected	**Healthy**
Real Classes	**Infected**	97.28 (TP)	2.72 (FP)
	Healthy	0 (FN)	100 (TN)

In order to see the effect of K value, the system is launched by selecting K=5. The detection result is shown in Table II.

TABLE II. Classification Results of Healthy and Infected Leaves (k = 5)

		predicted classes	
		infected	**Healthy**
Real Classes	**Infected**	97.33 (TP)	2.67 (FP)
	Healthy	0 (FN)	100 (TN)

The values shown in Tables I and II could be defined as below:

- *True Positive (TP)*: Infected leaves correctly identified as infected.
- *False Positive (FP)*: Healthy leaves incorrectly identified as infected.
- *True Negative (TN)*: Healthy leaves correctly identified as healthy.
- *False Negative (FN)*: Infected leaves incorrectly identified as healthy.

As shown in Table II, the TN indicator of the proposed method is 100%, meaning that all healthy leaves are predicted correctly, and the system does not have any wrong prediction among healthy leaves. The TP indicator for the proposed method is 97.33%, and FP is 2.67%, implying that among the infected leaves, only 2.67% of them are incorrectly predicted.

In addition to the above parameters, two other parameters can be defined as below:

- *Sensitivity*: Sensitivity, or true positive rate, refers to the system's ability to correctly predict infected leaves which do have the condition.

$$SENS=TP/(TP+FN) \quad (13)$$

- *Specifity*: Specificity, or true negative rate relates to the system's ability to correctly reject healthy leaves which do not have the condition.

$$SPEC=TN/(TN+FP) \quad (14)$$

From table II, the sensitivity and specificity of the proposed method can be obtained as 1 and 0.974, respectively. Also, overall average accuracy is achieved as 98.66%, which shows better results comparing to existing works such as [30].

VI. Conclusions and Future Work

Considering the importance of using machine vision methods to detect GLFV disease, this paper presents a method based on a combination of the FCM segmentation algorithm and the support vector machines (SVM) algorithm. Firstly, real images were collected in the Khorasan, Iran region. After applying some pre-processings, each image was divided into three segments using the FCM algorithm, and the percentage of the contaminated part to the healthy part for two first segments is considered as input to the SVM algorithm. The results of applying the SVM algorithm showed that the proposed algorithm is able to separate infected leaves from healthy leaves with an overall

accuracy of 98.6%. This paper showed that the proposed method for detecting diseases has good potential and is capable of detecting infected plants only by processing their leaves, which overcomes the existing limitations. Since other existing methods for detecting this virus are time-consuming and cost-effective, the use of this method can detect GFLV disease in time, and reduce the cost of diagnosis, which will decrease economic losses.

Considering the ease of preparation of images from any objects such as plants, trees, leaves etc., the proposed model has no limitation in distinguishing their different types, after learning suitably. Therefore, complete datasets in each way is desirable. Developing android or IOS based versions of the model could be very usefull for farmers.

References

[1] Kaur, Sukhvir, Shreelekha Pandey, and Shivani Goel. "Plants disease identification and classification through leaf images: A survey." *Archives of Computational Methods in Engineering* 26.2 (2019): 507-530.

[2] Al Bashish, Dheeb, Malik Braik, and Sulieman Bani-Ahmad. "Detection and classification of leaf diseases using K-means-based segmentation and." *Information Technology Journal* 10.2 (2011): 267-275.

[3] Mahmoudi, M. Khazaei, J., Vahdati, K., Taleb, M. "Identification of walnut diseases using visual machine techniques", first National Congress of Agricultural Science and Technology, Zanjan university, Iran, 2011, in Persian.

[4] Zhang, Shanwen, et al. "Cucumber leaf disease identification with global pooling dilated convolutional neural network." *Computers and Electronics in Agriculture* 162 (2019): 422-430.

[5] Sun, Xiaoxiao, et al. "Image Recognition of Tea Leaf Diseases Based on Convolutional Neural Network." *arXiv preprint arXiv:1901.02694* (2019).

[6] Shah, Nikhil, and Sarika Jain. "Detection of Disease in Cotton Leaf using Artificial Neural Network." *2019 Amity International Conference on Artificial Intelligence (AICAI)*. IEEE, 2019.

[7] HOSSEINI, H., ZAMANI D. MOHAMMAD, and A. ARBAB. "A recognition system to detect Powdery Mildew and Anthracnose fungal disease of cucumber leaf using image processing and artificial neural networks technique." (2018): 15-28.

[8] OMRANI, ELHAM, et al. "Identification of apple leaf varieties using image processing and adaptive neuro-fuzzy inference system." (2015): 67-75.

[9] Al-Hiary, Heba, et al. "Fast and accurate detection and classification of plant diseases." *International Journal of Computer Applications* 17.1 (2011): 31-38.

[10] Meunkaewjinda, A., et al. "Grape leaf disease detection from color imagery using hybrid intelligent system." *2008 5th international conference on electrical engineering/electronics, computer, telecommunications and information technology*. Vol. 1. IEEE, 2008.

[11] Wang, Haiguang, Guanlin Li, Zhanhong Ma, and Xiaolong Li. "Image recognition of plant diseases based on principal component analysis and neural networks." In 2012 8th International Conference on Natural Computation, pp. 246-251. IEEE, 2012.

[12] Dubey, Shiv Ram, et al. "Infected fruit part detection using K-means clustering segmentation technique." *International Journal of Interactive Multimedia and Artificial Intelligence* 2.2 (2013): 65-72.

[13] Belkhodja, Leila, and Djamila Hamdadou. "IMCAD: Computer Aided System for Breast Masses Detection based on Immune Recognition." *International Journal of Interactive Multimedia and Artificial Intelligence* 5.5 (2019): 97-108.

[14] Pujari, Devashish, Rajesh Yakkundimath, and Abdulmunaf S. Byadgi. "SVM and ANN based classification of plant diseases using feature reduction technique." *International Journal of Interactive Multimedia and Artificial Intelligence* 3.7 (2016): 6-14.

[15] Sutic, Dragoljub D., Richard E. Ford, and Malisa T. Tosic. *Handbook of plant virus diseases*. CRC Press, 1999. 553p.

[16] Vuittenz, A. 1970. Fanleaf of grapevine. In: N. W. Frazier. Virus disease of small fruits and grapevine. University of California, Berkeley, plant pathology. 217-228.

[17] Martelli G.P. 1993. Grapevine degeneration fanleaf. In: Martelli G.P. Graft transmissible diseases of grapevines. Handbook for detection and diagnosis, plant pathology. 9-18.

[18] Hewitt, W.B., Raski D.J., Goheen A.C., 1958. Nematode vector of soil-borne fanleaf virus of grapevines. Phytopathology 48: 586-595

[19] Hewitt, W. B., Martelli, G. P., Dias, H. F. & Taylor, R. H. 1970. Grapevine fanleaf virus. C. M. I/ A. A. B. Descriptions of Plant Viruses. No. 28.

[20] Z. Gholampur, M. Zaki. "Comparison of RNA Extraction Methods for the Identification of Grapevine fan leaf virus." *Journal of Plant Protection,*30(1):127-33. doi:10.22067/jpp.v30i1.40075.

[21] Nourinezhad, Zarghani sh., et al. "identification and detection of iranian isolates of grapevine fanleaf virus using green-grafting and RT-PCR." (2012): 381-391.

[22] Vuittenez, A. "Fanleaf of grapevine." Virus Disease of Small Fruits and Grapevine, University of California, Berkeley, plant pathology. (1970): 217-228.

[23] Fritsch, C., and M. Rudel. "Characterization and detection of grapevine fanleaf virus by using cDNA probes." *Phytopathology* 81 (1991): 559-565.

[24] John R. Crowther. Methods in Molecular Biology, The ELISA Guidebook. Second Edition. Humana Press, a part of Springer Science and Business Media, (2009).

[25] Lequin, Rudolf M. "Enzyme immunoassay (EIA)/enzyme-linked immunosorbent assay (ELISA)." Clinical chemistry 51, no. 12 (2005): 2415-2418.

[26] Rafiei, S., and P. Moradi. "Improving Performance of Fuzzy C-means Clustering Algorithm using Automatic Local Feature Weighting." (2016): 75-86.

[27] Zhi, Xiao-bin, Jiu-lun Fan, and Feng Zhao. "Robust local feature weighting hard c-means clustering algorithm." *Neurocomputing* 134 (2014): 20-29.

[28] Fatahi, Mazdak. 2015. A review on Support Vector Machines, Msc thesis, Razi university, (2015): 10, 1314, DOI: 10.13140/RG.2.1.4981.6487, in Persian.

[29] Wong, Tzu-Tsung. "Parametric methods for comparing the performance of two classification algorithms evaluated by k-fold cross validation on multiple data sets." *Pattern Recognition* 65(2017): 97-107.

[30] Gavhale, Kiran R., and Ujwalla Gawande. "An overview of the research on plant leaves disease detection using image processing techniques." *IOSR Journal of Computer Engineering (IOSR-JCE)* 16.1 (2014): 10-16.

Mojtaba Mohammadpoor

He has received his Bachelor of Engineering in Electrical Eng. (1998, Ferdowsi University of Mashhad, Iran), MSc in telecommunication Engineering (2006, Tehran Azad University, Iran), and Ph.D in telecommunication and network engineering (2012, University Putra Malaysia). He is currently an Assistant Professor in the Dept. of Electrical and Computer Eng., University of Gonabad, Iran. His research interests are network and network security, artificial intelligence, signal and image processing.

Mohadese Gerami Nooghabi

She has obtained Bachelor of Science (Agri.) in 2011 (Iran, Ferdowsi university of Mashad (FUM)), M.Sc. (Agri. Plant pathology) in 2014 (Iran, FUM) and she is Currently plant virology Ph.D student at FUM.

Zahra Ahmedi

She has obtained her Bachelor of Engineering in computer engineering in 2017 from University of Gonabad, Iran.

Classification-based Deep Neural Network Architecture for Collaborative Filtering Recommender Systems

Jesús Bobadilla*, Fernando Ortega, Abraham Gutiérrez, Santiago Alonso

Universidad Politécnica de Madrid, Carretera de Valencia Km 7, 28031 Madrid (Spain)

Received 17 December 2019 | Accepted 31 January 2020 | Published 18 February 2020

Abstract

This paper proposes a scalable and original classification-based deep neural architecture. Its collaborative filtering approach can be generalized to most of the existing recommender systems, since it just operates on the ratings dataset. The learning process is based on the binary relevant/non-relevant vote and the binary voted/non-voted item information. This data reduction provides a new level of abstraction and it makes possible to design the classification-based architecture. In addition to the original architecture, its prediction process has a novel approach: it does not need to make a large number of predictions to get recommendations. Instead to run forward the neural network for each prediction, our approach runs forward the neural network just once to get a set of probabilities in its categorical output layer. The proposed neural architecture has been tested by using the MovieLens and FilmTrust datasets. A state-of-the-art baseline that outperforms current competitive approaches has been used. Results show a competitive recommendation quality and an interesting quality improvement on large number of recommendations, consistent with the architecture design. The architecture originality makes it possible to address a broad range of future works.

Keywords

Deep Learning, Neural Classification, Neural Collaborative Filtering, Recommender Systems, Scalable Neural Architecture.

DOI: 10.9781/ijimai.2020.02.006

I. Introduction

RECOMMENDER Systems (RS) [1] play an important role to address the information overload in Internet. RS can use diverse sources of information: votes from users to items, purchased products or consumed services, social information, geographical coordinates, demographic information, etc. The most accurate RS are the Collaborative Filtering-based (CF) ones; they learn from all the existing implicit or explicit information about how users vote or consume items. It is usual to merge RS types (hybrid RS) [2] to improve accuracy results. CF RS can be reinforced by means of demographic [3], context-aware [4], content-based [5] and social [6] information. The scope of the RS has extended and currently covers an endless number of targets: tourism [7], films [8], networks [6], restaurants [9], e-learning, fashion [10], news [11], healthcare [12], etc.

CF RS kernels have been implemented by using machine learning methods: memory-based and model-based ones. KNN was the main memory-based method, but currently model-based algorithms are used due to their accuracy superiority. Matrix Factorization (MF) is the most implemented approach, since it provides accurate recommendations, it is easy to understand, and it obtains a good performance. There are several MF variations such as PMF [13] [14], BNMF [15], BPR [16] and eALS [17]. MF extracts the complex relations between items and users and codes them into a reduced number of hidden factors. Predictions are obtained by making the dot product of users and items. A MF drawback is the linearity of the dot product: it does not allow to accurately combine hidden factors to provide the most suitable predictions.

RS research is heading towards solutions based on deep learning [18] [19]. Some of the approaches make use of non-collaborative data: images information [9] [10], text [11] [20], music [21] [22], videos [23] [24], session ID [25] [26], etc. Depending on the type of data, different deep learning architectures are used: CNN [27] [28], RNN [29] [30], MLP [31] [4], Autoencoder [8] [32], etc. The previous examples implement specific content-based approaches or RS hybrid solutions. The CF RS core is based on the ratings to items casted by users. The deep learning solutions to this challenge can be classified as: a) Neural Collaborative Filtering (NCF) [33], and b) Deep Factorization Machines (deepFM) [34]. NCF are usually based on dual neural networks to simultaneously process users and items information. The two NCF references are: 1) Neural Network Matrix Factorization (NNMF) [35] and Neural Collaborative Filtering (NCF) [33].

NCF reports better performance than NNMF, so we put the focus on it. Fig. 1 shows the NCF architecture; as explained, it takes the sparse user and item raw data vectors. Above the sparse layer is the fully connected embedding layer that projects the sparse representation to a dense vector. Then a Multilayer Perceptron (MLP) combines the two pathways features by concatenating them.

A deep learning type of architecture used in CF combines wide & deep learning [36] [37]; the wide component is a single-layer perceptron, whereas the deep component is a MLP. Combining the two learning techniques enables the RS to capture both generalization

* Corresponding author.
E-mail address: jesus.bobadilla@upm.es

and memorization. The wide component captures simple data relations (memorization), whereas the deep component can make more valuable abstractions (generalization). Our proposed method can be extended to the wide & deep learning architecture and it will be recommended as future work. The DeepFM architecture joins a wide and a deep component to share the same input raw feature vector. It allows to simultaneously learn low and high order feature interactions from the input raw features. DeepFM consists of two blocks: FM component and deep component. The FM component is a factorization machine [38] to learn feature interactions. DeepFM achieves a 0.48% (Company dataset) and a 0.33% (Criteo dataset) of accuracy gain compared to NCF. This is a small performance improvement, so we will use the best known NCF as baseline.

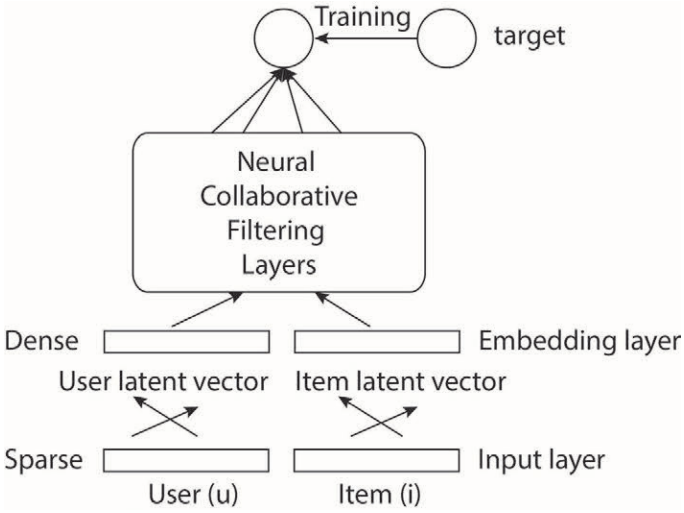


Fig. 1. NCF baseline architecture.

Classical methods make use of MF to make predictions. Some neural collaborative filtering methods replace the MF linear dot product used to predict for a non-linear MLP stage. Neural network methods that do not use MF often are based on deep hybrid models: CNN, RNN, etc. that exploit additional data to the rating matrix, usually not available for most of the existing RS. Wide and multi-view deep learning architectures have been used as pure neural network RS approaches; their main drawback is the size of the wide layers that jeopardize scalability. The proposed method architecture is scalable, because it is just based on the RS item dimensionality, relying the huge RS user dimensionality to the number of samples used to train the model.

As we have seen, existing RS deep neural approaches make use of a MF level or an embedding layer to catch the item and user latent vectors. They are regression-based architectures: for each user, recommendations are chosen from its N best predictions. Then, to recommend a user it is necessary to run I predictions (I is the number of items in the RS). Our proposed deep neural architecture does not need an external machine learning level, as the MF one. It does not use, either, explicit embedding layers. Its original architecture comes from the fact that it makes recommendations by means of a classification process instead a regression one. The classification approach requires a different design that the existing ones: it will learn by using categorical labels instead numerical values. Additionally, we are committed to create a simple and scalable architecture. To meet the stated aims we have decided to renounce to the numerical rating values and predictions: we will only make use of the binary relevant/non-relevant rating and the binary voted/non-voted information. As the reader can imagine, this decision opens a new range of architectural choices. Of course, a key question must be answered: will the accuracy of recommendations decrease significantly? State of art papers indicate that accuracy is not significantly affected, although it could vary depending on the dataset. From [39], the graph in Fig. 2 is provided.

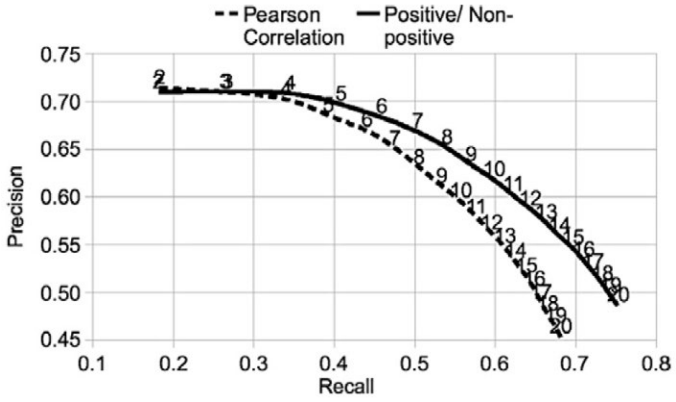


Fig. 2. MovieLens 1M Precision/Recall obtained by transforming all 4 and 5 votes into relevant votes, and 1, 2 and 3 votes into non-relevant votes, compared to the results obtained using the numerical values [39].

We will call our proposed method as Neural Classification-based Collaborative Filtering (NCCF), and we are going to motivate this approach by explaining, with the help of representative figures: a) Its innovative design, b) Its scalability basis, and c) Its straightforward method to get predictions of each user. Fig. 3 shows the evolution from MF methods to the proposed NCCF. First of all, both the MF and the NCF models (Fig. 3a & 3b) are regression-based: they provide prediction values, and the RS recommends the highest N predictions. Instead, NCCF (Fig. 3c) is based on a classification neural network. Furthermore, NCCF is not based on a previous machine learning MF process or an embedding layer; instead, raw rating data is used avoiding the necessity of feature engineering stages [18]. It is important to highlight that NCCF raw data is made up by user vectors, where each vector contains items information. NCCF does not combine user vectors with item vectors, such as the NCF baseline does. It does not combine, either, user/item vectors with dense embedding coming from a factorization machine. This make the proposed approach simpler and more scalable than the current baselines.

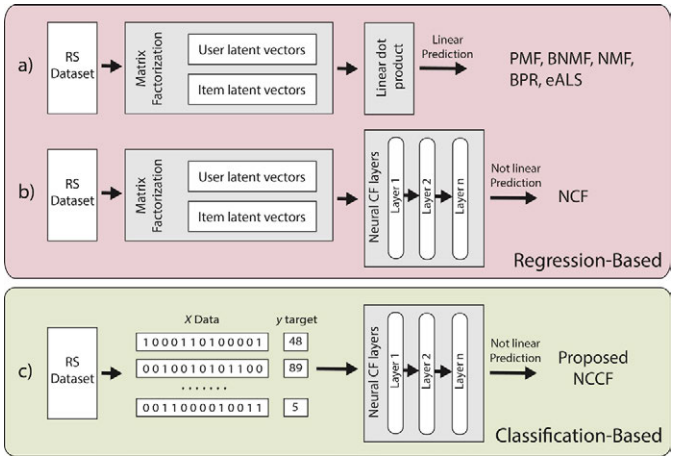


Fig. 3. Comparative between the proposed method c) and the baseline ones: a) & b).

To improve understanding of the proposed method advantages, Fig. 4 shows an outline involving the learning parameters of both the proposed and the baseline models. The MF machine learning algorithm needs to learn a set of F factors for each user (U) and for each item (I) in the dataset. F usually ranges from 12 to 40 and, as shown in Fig. 4a,

in commercial RS the number of users is much larger than the number of items. The necessary MF parameters is: (U+I)F. NCF approaches are built on MF, so Fig. 4b reproduces the MF parameters shown in Fig. 4a. Additionally, the MLP needs learning parameters; if the first inner layer contains n neurons (Ln), we need (U+I+Ln)F to run the MF model and to process the first MLP layer. The following layers will need more learning parameters (Lm), but usually Lm (number of neurons in the following layer) is not a high number. The output layer just needs Lm parameters, since we just need a neuron to obtain the regression value (the prediction). Overall, the NCF baseline, when the MLP has two inner layers, needs (U+I+Ln)F+(Ln+1)Lm parameters. The NCCF proposed method is designed to be scalable: it makes use of a tiny portion of the necessary parameters of the NCF baseline. Even although it uses an output layer of dimension I, it just needs I(Lm+Ln)+LmLn learning parameters.

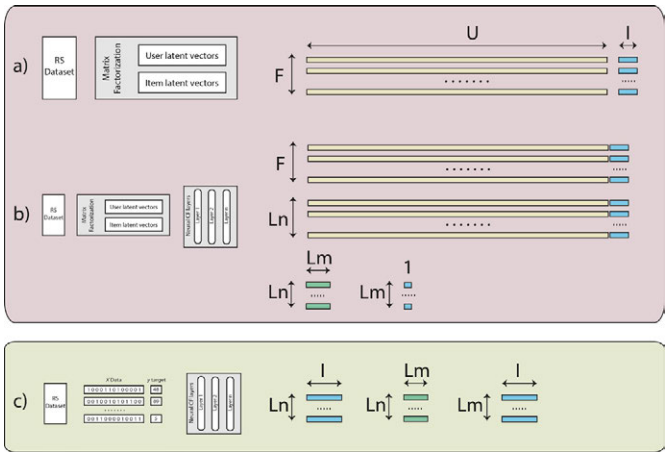


Fig. 4. Involved learning parameters in both the proposed c) and the baseline models: a) & b).

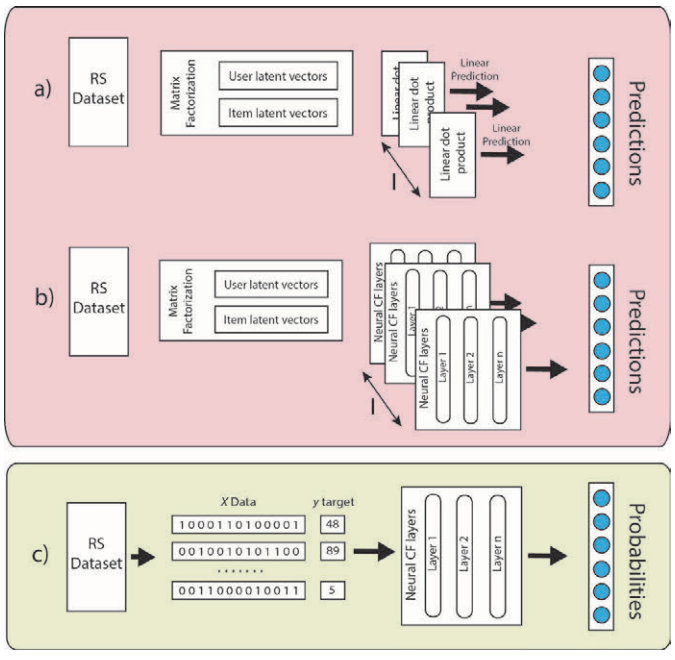


Fig. 5. Process to predict items values for each user in both the proposed c) and the baseline models: a) & b).

Another relevant improvement of the proposed method is its fully neural network processing: it allows to model the non-linear relations from the raw data to the result (predictions). It does not start from previous machine learning results that could have lost some of the rich data dependencies. Furthermore, NCCF processes RS predictions in a different way that the baselines do, as it is shown in Fig. 5: in order to provide N recommendations to a user, the MF approach (Fig. 5a) needs to run I dot products (if we dismiss the items voted for the user). The NCF baseline (Fig. 5b) must run forward I times its MLP: each one providing the pair <user,item> to the neural network. Instead, to provide recommendations to a user, the proposed NCCF (Fig. 5c) just have to run forward once its MLP. Since we are working with a categorical target, the output layer will provide I classification probabilities. To make N recommendations we just have to select the N highest probabilities; this is a simple and straightforward operation.

II. Proposed Method

This section explains the proposed methods both in a conceptual way and in its formalized version. Each explanation type is made in a separated subsection. First subsection is centered in the concepts and the motivation of each proposed approach; a data-toy running example is provided as well as the proposed architecture explanation through a representative figure. Second subsection provides the proposed methods formalization to facilitate their implementation; formalization is explained by means of a data-toy running example.

A. Concepts

We propose two related methods: a) the short one, and b) the long one. The short method is based on the relevant ratings of the dataset: ratings that exceed the value of an established threshold (e.g.: 4 stars in a RS where users can vote from 1 to 5 stars). We will lose the numerical fine grain of the votes and we renounce to make numerical predictions (e.g.: we recommend you this item with the 4.2 stars value). We will not work with the usual rating data; instead we make an abstraction by selecting the binary relevant/non-relevant information. Table I shows a data-toy example of CF RS dataset ("Original dataset") and the abstracted relevant vs. non-relevant information ("Proposed short and long methods preliminary information"). Our aim is to be able to accurately recommend RS items without making use of the usual numerical prediction that in fact is a regression process. We will make RS recommendation by means of a classification process, and the proposed classification process will be mainly based on the rating relevancy abstraction.

TABLE I. Preliminary Information

Original dataset						
rating	i0	i1	i2	i3	i4	i5
u0	•	4	•	5	2	•
u1	3	1	5	4	•	5
u2	1	•	4	•	•	4
Proposed short and long methods preliminary information						
relev.	i0	i1	i2	i3	i4	i5
u0	0	1	0	1	0	0
u1	0	0	1	1	0	1
u2	0	0	1	0	0	1
Proposed long method preliminary information						
relev.	i0	i1	i2	i3	i4	i5
u0	0	1	0	1	0	0
u1	0	0	1	1	0	1
u2	0	0	1	0	0	1
voted	i0	i1	i2	i3	i4	i5
u0	0	1	0	1	1	0
u1	1	1	1	1	0	1
u2	1	0	1	0	0	1

Preliminary information for both the short and long proposed methods. The value "•" means non-voted item. Threshold = 4.

By converting ratings to relevant/non-relevant data we lose a type of information: voted or non-voted item. This is due to the inherent sparse nature of CF RS: users can only vote a reduced number of the available items because we ca not see all the films, buy all the products or consume all the services that a modern RS provides. Our proposed long method incorporates the voted/non-voted information and adds it to the short method information; Table I ("Proposed long method preliminary information") shows the concept. Neural networks are able to relate both types of information to make their job (in this case to accurately classify); this means that using the long model it is possible to discern from: a) relevant vote, b) non-relevant vote, and c) not voted. The short method approach works without the 'not voted' information.

The long model approach has the advantage of working with more information than the short model, and it is expected to provide better results. Its main drawback is the size of its necessary input and output neural network layers: it will need twice input and output neurons than the short method approach (since it provides two types of information: voted/non-voted and relevant/non-relevant). The key issue is the obtained balance between performance and accuracy. Accuracy mainly will come from the relevant/non-relevant information, whereas voted/non-voted data can improve results. While the "voted" items information meaning is clear, the "non-voted" items information is a controversial subject: non-voted items meanings can be: a) the user does not know the item, b) the user knows the item but he/she does not know if he/she will like it, c) the user knows that he/she does not like the item, but he/she did not vote it, and d) the user knows that he/she likes the item, but he/she did not vote it. RS aim is centered in the a) and b) options. Our short method loses the semantic difference between not to vote an item and to vote this item as non-relevant. The lost information will be more or less important depending on the relative significance of cases c) and d). The proposed short method assigns the meaning "voted as not relevant" to the non-voted items, so case c) does not affect to the proposed short model accuracy. The proposed long method does not assign a "negative" semantic to the non-voted items, as the short model does; this means that the long method correctly manages the case d), whereas the short method does not. The accuracy difference between the proposed methods rely on the significance of the case d) in each RS: if the users are more active, this case importance decreases and vice versa. Published studies show that using binary relevancy information instead numerical votes does not significantly change accuracy results [39].

On the basis of the preceding information (Table I) both proposed methods create a set of samples to train and test the classification-based RS. For each relevant vote (numbers 1 in the relevant sections of Table I) we create a sample: data and target, thus the model number of samples will be the existing number of relevant ratings in the dataset. Each dataset user will generate a number of samples equal to her number of relevant ratings. *The key concept is to train our model with the relevant ratings of each user except the first one (this is the X data sample), and to use this rating as a target (this is the y label of the sample). This process is repeated until the last relevant rating of the user is reached*. As an example, the obtained samples from the sort method information in Table I are shown in Table II.

TABLE II. X Data and Y Label Samples from Table I; Short Method

X	i0	i1	i2	i3	i4	i5		y	
$X_{0,1}$	0	0	0	1	0	0		$Y_{0,1}$	1
$X_{0,3}$	0	1	0	0	0	0		$Y_{0,3}$	3
$X_{1,2}$	0	0	0	1	0	1		$Y_{1,2}$	2
$X_{1,3}$	0	0	1	0	0	1		$Y_{1,3}$	3
$X_{1,5}$	0	0	1	1	0	0		$Y_{1,5}$	5
$X_{2,2}$	0	0	0	0	0	1		$Y_{2,2}$	2
$X_{2,5}$	0	0	1	0	0	0		$Y_{2,5}$	5

Our proposed methods provide valuable information to make a RS classification: each user relevant rating can be predicted from the set of the remaining relevant ratings from this user. Additionally, each user relevant rating can be predicted from the rest of users' data. In Table II we can observe how the model can learn that target 3 (y_0,3 and y_1,3) can be predicted when i1 is relevant or when i2 and i5 are relevant. In the same way, target 5 prediction will be particularly probable when i2 is relevant, and this is an information coming from users 1 and 2. Current datasets contain hundreds of thousands or millions relevant ratings, thus the proposed methods will provide rich models to be used in artificial intelligence classification algorithms.

A fundamental decision design in the proposed methods is to maintain the dimensionality of the samples in reasonable sizes; we want: a) A large number of samples to train and test the model, and b) A reasonable size of the samples in the model in order to maintain the performance (time consuming) of the classification algorithms. Since commercial RS have much more users than items, to make these requirements compatible, we must use the RS user dimension to generate samples, and the RS item dimension to stablish the samples size. Some RS have a large number of items; in such cases, the proposed short model can be particularly appropriated since it reduces by half the samples size.

Fig. 6 shows a generic neural network model implementing these paper's proposed methods. Each input data column (on the left) represents a sample. Our proposed methods provide the samples; e.g.: each row in Table II feeds the corresponding column sample in the proposed short method neural network architecture. The proposed long method adds the voted/non-voted information to the neural network. We also provide the categorical labels, necessary for the classification task: there are as many classes as items in the RS (I number of items in Fig. 6). Once the neural network has been trained, for each new input sample (new user) the neural network predicts a set of probabilities: the probability of each class (item) to be classified from the sample data (user). We just have to take the N highest probabilities to obtain the N recommendations; this process is drawn on the right of both methods in Fig. 6.

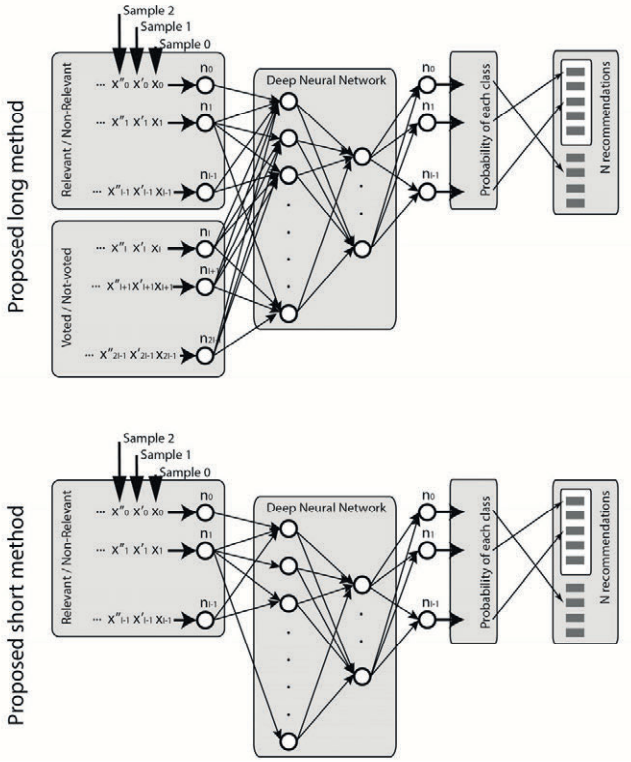


Fig. 6. Proposed neural network architecture for both the short and long methods.

B. Formalization and Data-Toy Example

Let I be the set of the RS items.

Let U be the set of the RS users.

Let $V = \{1, 2, 3, 4, 5, \bullet\}$ be the set of available ratings. • means empty rating.

Let R be the set of the dataset ratings:

$$R = \{< u, i, v > \ \forall i \in I, \forall u \in U, v \in V\} \quad (1)$$

Let R_u be the set of the dataset ratings casted by user u:

$$R_u = \{< i, v > \ \forall i \in I, v \in V\} \quad (2)$$

Let $R_{u,i}$ be the rating casted by user u to the item i:

$$R_{u,i} = v \Leftrightarrow < u, i, v > \in R_u \quad (3)$$

Let I' be an ordered list of the I elements.

Let $\theta \in V - \{\bullet\}$ be the rating relevance threshold (4)

Let D_u be the intermediate data for the proposed method short version. D_u differentiates the relevant and not relevant ratings casted by user u.

$$D_u = \{< u, i, 1 > \ \forall i \in I' \mid R_{u,i} \neq \bullet \ and \ R_{u,i} \geq \theta\} \cup \{< u, i, 0 > \mid R_{u,i} = \bullet \ or \ (R_{u,i} \neq \bullet \ and \ R_{u,i} < \theta)\} \quad (5)$$

where $D_{u,i}$ is the relevance of the $R_{u,i}$ vote:

$$D_{u,i} = c \Leftrightarrow < u, i, c > \in D_u \quad (6)$$

Let $X_{u,s}$ be the sample data s from D_u, and $Y_{u,s}$ the target data s from D_u:

$$X_{u,s} = \{D_u - < u, s, 1 >\} \cup \{< u, s, 0 >\}, \forall s \in I', \mid D_{u,s} = 1 \quad (7)$$

$$y_{u,s} = s \ \Leftrightarrow < u, s, 1 > \in \ X_{u,s} \quad (8)$$

Finally, the X and Y data to train the proposed short method is:

$$X = \{X_{u,s}: \forall u \in U, \forall s \in I'\} \quad (9)$$

$$Y = \{y_{u,s}: \forall u \in U, \forall s \in I'\} \quad (10)$$

Let $D_u || D'_u$ be the intermediate data for the proposed method long version. D'_u differentiates the user u voted and not voted items.

$$D'_u = \{< u, i, 1 > \ \forall i \in I' \mid R_{u,i} \neq \bullet\} \cup \{< u, i, 0 > \ \forall i \in I' \mid R_{u,i} = \bullet\} \quad (11)$$

$$D'_{u,i} = i \Leftrightarrow < u, i, 1 > \in D'_u \quad (12)$$

Let $X'_{u,s}$ be the sample data s from D'_u , and $y'_{u,s}$ the target data s from D'_u:

$$X'_{u,s} = \{D'_u - < u, s, 1 >\} \cup \{< u, s, 0 >\}, \ \forall s \in I' \mid D'_{u,s} = 1 \quad (13)$$

$$y'_{u,s} = \ s \Leftrightarrow < u, s, 1 > \in X'_{u,s} \quad (14)$$

Finally, the X' and Y' data to train the proposed short method is:

$$X' = \{X_{u,s}: \forall u \in U, \forall s \in I\} \quad (15)$$

$$Y' = \{y_{u,s}: \forall u \in U, \forall s \in I\} \quad (16)$$

Once the model has been trained by using the X and Y sets or the X' and Y' sets, we obtain a result set W with cardinality I. W contains the probability of each item to be recommended. These results are predicted by using a neural network containing I "softmax" output neurons. The N expected recommendations Z will be the N highest results from W.

$$Z = \{w_i \in W \mid w_i \geq w_j, w_j \in Z^c\}, \ \#Z \ = \mathrm{N} \quad (17)$$

We will explain the previous equations by means of the data-toy example on Table I. It shows the ratings casted by users 0, 1 and 2 on items 0 to 5. The • symbol means not voted. This data-toy uses the V set definition from the above equations

From the data-toy, the sets and lists from previous formalization are:

$$R = \{< 0,0,\bullet >, < 0,1,4 >, < 0,2,\bullet >, < 0,3,5 > \ldots < 2,4,\bullet >, < 2,5,4 >\} \quad (1)$$

The number that accompanies each expression refers to the applied equation in previous formalization.

The set of the dataset ratings casted by users 0, 1 and 2:

$$R_0 = \{< 0,0,\bullet >, < 0,1,4 >, < 0,2,\bullet >, < 0,3,5 >, < 0,4,2 >, < 0,5,\bullet >\} \quad (2)$$

$$R_1 = \{< 1,0,3 >, < 1,1,1 >, < 1,2,5 >, < 1,3,4 >, < 1,4,\bullet >, < 1,5,5 >\} \quad (2)$$

$$R_2 = \{< 2,0,1 >, < 2,1,\bullet >, < 2,2,4 >, < 2,3,\bullet >, < 2,4,\bullet >, < 2,5,4 >\} \quad (2)$$

As an example, we can see that:

$$R_{0,0} = \bullet, R_{0,1} = \ 4, R_{1,2} = \ 5, R_{1,5} = \ 5, R_{2,2} = \ 4, \text{etc.} \quad (3)$$

We will stablish the $\theta = 4$ threshold (4)

The set of information for the training and testing short proposed method is:

$$D_0 = \{< 0,0,0 >, < 0,1,1 >, < 0,2,0 >, < 0,3,1 >, < 0,4,0 >, < 0,5,0 >\} \quad (5)$$

$$D_1 = \{< 1,0,0 >, < 1,1,0 >, < 1,2,1 >, < 1,3,1 >, < 1,4,0 >, < 1,5,1 >\} \quad (5)$$

$$D_2 = \{< 2,0,0 >, < 2,1,0 >, < 2,2,1 >, < 2,3,0 >, < 2,4,0 >, < 2,5,1 >\} \quad (5)$$

The X data and the y targets to the classification neural network is extracted from D_u. Each n_i column feeds a neuron of the neural network input layer. Each u_i row is a sample for the training or the testing process.

From equations (7) to (10) we obtain the input X and the output Y values to feed and train the neural network. This is the proposed short method to arrange data in order to make RS classification and then RS recommendation. shows the obtained results. As it can be seen, the method provides the categorical values for the labels.

The additional set of information for the training and testing long proposed method tell us about the voted and not voted items:

$$D'_0 = \{< 0,0,0 >, < 0,1,1 >, < 0,2,0 >, < 0,3,1 >, < 0,4,1 >, < 0,5,0 >\} \quad (11)$$

$$D'_1 = \{< 1,0,1 >, < 1,1,1 >, < 1,2,1 >, < 1,3,1 >, < 1,4,0 >, < 1,5,1 >\} \quad (11)$$

$$D'_2 = \{< 2,0,1 >, < 2,1,0 >, < 2,2,1 >, < 2,3,0 >, < 2,4,0 >, < 2,5,1 >\} \quad (11)$$

The proposed long method concatenates both D_u and D'_u: $D_u || D'_u$ (Table IV).

From equations (13) to (16) we obtain the input X' and output Y' values to feed the neural network training. This is the proposed long model to arrange data in order to make RS classification and then RS recommendation. We can see it in Table V. The categorical output values are the ones shown in Table III.

TABLE III. DATA-TOY EXAMPLE: CLASSIFICATION DATA AND CATEGORICAL LABELS FOR THE PROPOSED SHORT METHOD

X	Input neurons					
$X_{u,s}$	n_0	n_1	n_2	n_3	n_4	n_5
$X_{0,1}$	0	0	0	1	0	0
$X_{0,3}$	0	1	0	0	0	0
$X_{1,2}$	0	0	0	1	0	1
$X_{1,3}$	0	0	1	0	0	1
$X_{1,5}$	0	0	1	1	0	0
$X_{2,2}$	0	0	0	0	0	1
$X_{2,5}$	0	0	1	0	0	0

y	Output neurons					
$y_{u,s}$	n_0	n_1	n_2	n_3	n_4	n_5
$y_{0,1}$	0	1	0	0	0	0
$y_{0,3}$	0	0	0	1	0	0
$y_{1,2}$	0	0	1	0	0	0
$y_{1,3}$	0	0	0	1	0	0
$y_{1,5}$	0	0	0	0	0	1
$y_{2,2}$	0	0	1	0	0	0
$y_{2,5}$	0	0	0	0	0	1

TABLE IV. DATA-TOY EXAMPLE: VOTED/NON-VOTED INFORMATION

	n_0	n_1	n_2	n_3	n_4	n_5	n_0	n_1	n_2	n_3	n_4	n_5
u_0	0	1	0	1	0	0	0	1	0	1	1	0
u_1	0	0	1	1	0	1	1	1	1	1	0	1
u_2	0	0	1	0	0	1	1	0	1	0	0	1

TABLE V. DATA-TOY EXAMPLE: CLASSIFICATION DATA AND CATEGORICAL LABELS FOR THE PROPOSED LONG METHOD

X'	Input neurons											
	Relevant vs. non-relev.						Voted vs. non-voted					
$X_{0,s}$	n_0	n_1	n_2	n_3	n_4	n_5	n_0	n_1	n_2	n_3	n_4	n_5
$X_{0,1}$	0	0	0	1	0	0	0	0	0	1	1	0
$X_{0,3}$	0	1	0	0	0	0	0	1	0	0	1	0
$X_{1,2}$	0	0	0	1	0	1	1	1	0	1	0	1
$X_{1,3}$	0	0	1	0	0	1	1	1	1	0	0	1
$X_{1,5}$	0	0	1	1	0	0	1	1	1	1	0	0
$X_{2,2}$	0	0	0	0	0	1	1	0	0	0	0	1
$X_{2,5}$	0	0	1	0	0	0	1	0	1	0	0	0

III. EXPERIMENTS AND RESULTS

The designed experiments to test our proposed approaches make use of the MovieLens 1M[1] [45] dataset and the FilmTrust [44] dataset. The tested classification quality measures will mainly be the Precision and Recall ones [1]. The chosen relevancy threshold has been 4 for MovieLens and 3 for FilmTrust. The tested number of recommendations N has ranged from 1 to 96, in incrementes 5. For both datasets, the training set has been randomly obtained taking the 80% of the samples, whereas the test set has used the remaining 20%. Several neural network architectures have been tested and we have chosen the ones that have achieved better results. The Keras[2] deep learning library has been used to run experiments. For experiments reproducibility purposes we openly provide the necessary files to feed the neural networks[3]; these files have been obtained by implementing the proposed methods.

Experiment explanations have been organized by using two subsections. First subsection is devoted to compare the proposed long approach versus the proposed short approach. These experiments have been run using the MovieLens dataset, and they are explained by means of illustrative architectural figures. Results are provided both for classification accuracy and for recommendation accuracy. Second subsection compares the proposed short method with a representative state of art deep learning baseline: NCF. Recommendation and classification results are discussed both for MovieLens and FilmTrust datasets.

A. Long versus Short Approaches Comparative

Our first set of experiments compares the results of the proposed short method versus the proposed long method, making use of the MovieLens dataset. The number I of items is 1682 in this dataset. We have trained our proposed long method by using a dense deep neural network involving a 2 x 1682 = 3364 size for the input layer, 1682 size in the output layer, 400 neurons in the first internal layer, followed by a 0.2 dropout layer, and finally 200 neurons in the second internal layer, followed by another 0.2 dropout layer. All the layers use 'relu' activation, except the output one that uses 'softmax' in order to provide probabilistic categorical results. The chosen loss function has been 'categorical cross entropy' since we are using categorical labels. The selected optimizer: 'adam'. Fig. 7 shows the explained neural network architecture. The resulting number of trainable parameters is 1,308,632.

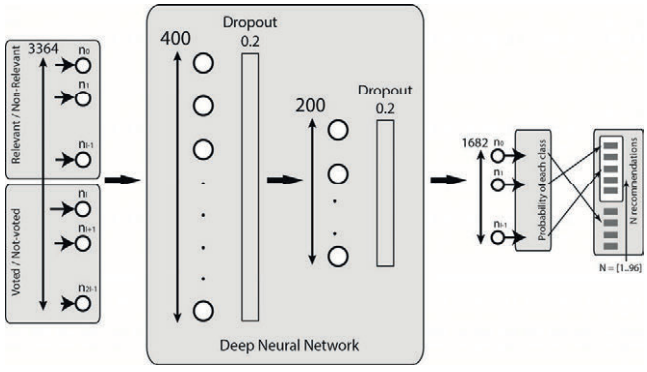


Fig. 7. Neural network architecture used to train the proposed long method on MovieLens 1M dataset.

Fig. 8 shows the results of the Fig. 7 training and testing processes. As it can be seen there is not overfitting, the loss function decreases to very low values and the accuracy raises to reach a 0.84 value. The

[1] https://movielens.org/

[2] https://keras.io/

[3] http://rs.etsisi.upm.es/

neural network learns in just 80 epochs. These results show good expectations to get accurate precision and recall values.

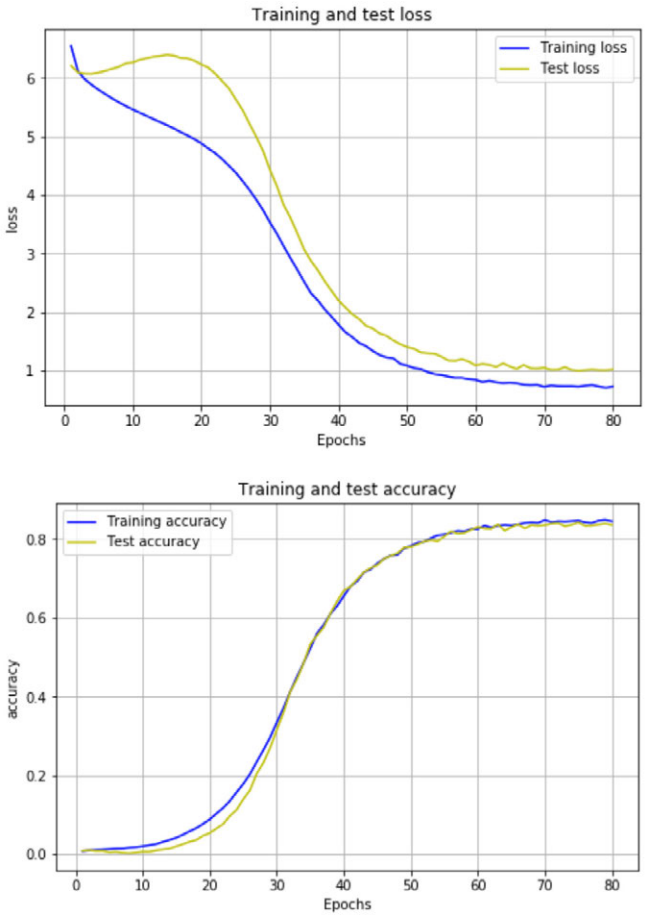


Fig. 8. The proposed long method results obtained by training the Fig. 7 deep neural network and using the MovieLens 1M dataset. Loss function (top graph) and accuracy (bottom graph) results both for training and test sets.

Analogously to the performed experiment using the proposed long model, an additional experiment has been made using the proposed short model. In this case, the size of the neural network input layer is the number of items of the dataset: 1682. Since the input layer size is half of the previous experiment one, we have reduced the neural network internal layer sizes to 200 and 100 and we have maintained the rest of parameters: two 0.2 dropouts, 'relu' and 'softmax' activations, etc. The resulting number of trainable parameters is 526,582. Fig. 9 shows the designed neural network architecture, and in Fig. 10 we can see that the obtained loss and accuracy values and evolutions are similar to the proposed long method ones. This case also feeds the expectation of accurate precision and recall results from recommendations.

It is important to distinguish between the loss and accuracy results, on one side, and the precision and recall, on the other side. Loss and accuracy are related to the training and testing results of the deep neural network (biggest square in Figs. 7 and 9). Precision and recall are related to the recommended items (most-right square in Figs. 7 and 9). Low loss values usually provide high accuracy ones. High accuracy values, in our context, means that we are correctly classifying items. Even if we classify correctly an item it does not directly mean that we have got a correct recommendation: probably this item has been voted by the user and it cannot be recommended, or it is possible that we have reached our limit of N recommendations and we cannot recommend the item.

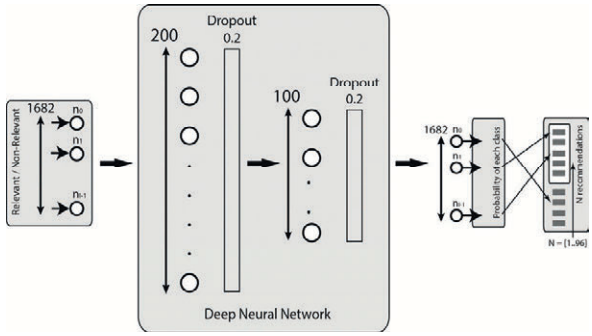


Fig. 9. Neural network architecture used to train the proposed short method. MovieLens 1M dataset.

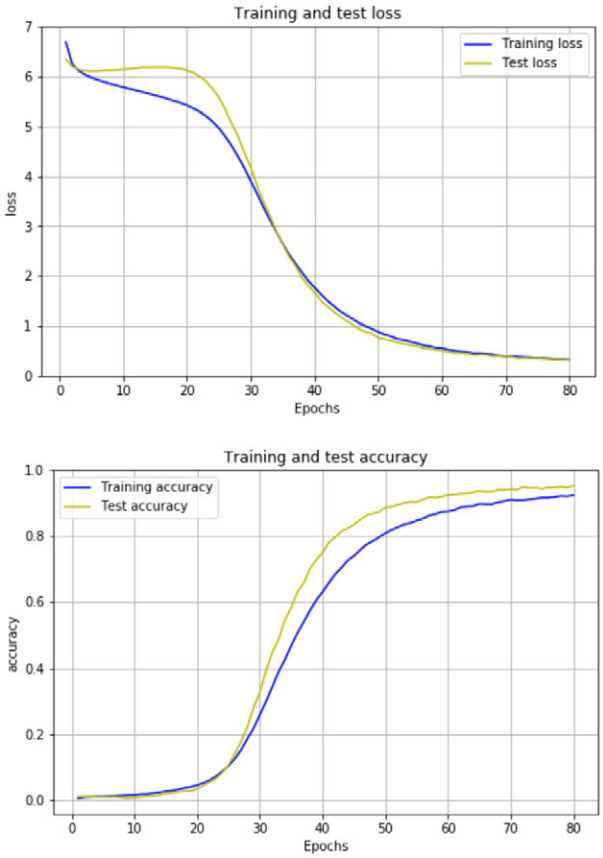


Fig. 10. The proposed short method results obtained by training the Fig. 9 deep neural network and using the MovieLens 1M dataset. Loss function (top graph) and accuracy (bottom graph) results both for training and test sets.

From each sample to predict (set of votes from some user) we will get up to N recommended items. These are the ones with the highest output probabilities (right boxes in Figs. 7 and 9; equation 17 in the formalization section). From these recommended items we get the combined precision, recall: F1, quality values using the classical information retrieval approach. Comparative results of both proposed methods are shown in Fig. 11; we can see a better performance of the short method when N is low, whereas for high number of recommendations the long method quality is something better. It makes sense, since the higher the N the more information is needed. The key issue here is that the quality difference between both approaches is not proportionally significant, whereas the deep neural network size needed to implement the long method is substantially bigger than the one needed to implement the short method (Figs. 7 and 9). Accordingly, we select the proposed short method as the preferred one, and we will use it in the rest of this paper's experiments.

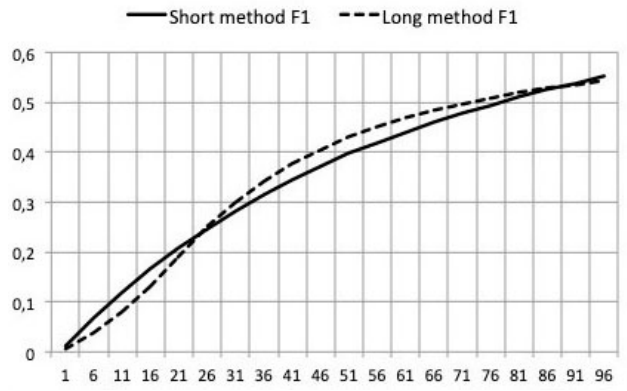


Fig. 11. Comparative recommendation F1 results obtained from RS classification using both the proposed short and long methods. Dataset: MovieLens 1M. y-axis: Combined precision & recall (F1); x-axis: Number of recommendations N.

B. The Proposed Approach Performance

In this section, experiments are designed to compare our chosen approach (the proposed short method) with a representative state of art deep learning baseline: NCF. We have called our proposal as: NCCF (Neural Classification-based Collaborative Filtering). Experiments are run on the MovieLens and FilmTrust datasets. Precision and Recall quality measures have been tested. The chosen relevancy thresholds are 4 for MovieLens and 3 for FilmTrust. The tested number of recommendations N has ranged from 1 to 96, step 5. For both datasets, the training and test samples have been randomly split in 80% and 20% set sizes.

Experiments make use of the state of art baseline NCF [33]. NCF (Neural-based Collaborative Filtering) is a neural architecture that can express and generalize Matrix Factorization. Its design is shown in Fig. 12. NCF has been tested using several current baselines: BPR [16] and eALS [17] that are optimizations of the Matrix Factorization model, and ItemKNN [40]: the standard item-based KNN. The chosen NCF baseline can be classified as Neural Collaborative Filtering [18] and it provides similar accuracy to deepFM [34]; NCF integrates matrix factorization and Multilayer Perceptrons (MLP). Their main advantages are: a) Its performance, b) It avoid tedious feature engineering, and c) It does not need additional information to the CF dataset ratings. NCF is considered as a robust and accurate deep learning design for CF RS. The main current alternatives to the NCF are: a) The autoencoder based ones, such as CDL [41], and b) The deep hybrid models for recommendation [42] [43] [11]. Autoencoder based approaches avoid the non-neural network stage (MF) to obtain features from data. They increase the neural network complexity without reaching a remarkable accuracy rise. Deep hybrid models are based on additional data to the dataset ratings: demographic, social, context-aware, content-based information, etc. They do not provide a universal method to CF RS.

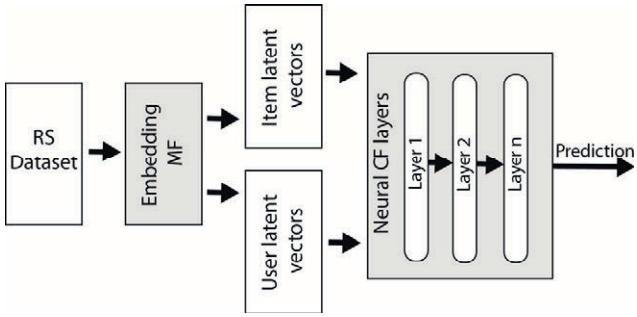


Fig. 12. NCF architecture used as baseline.

Once the baseline choice has been motivated, we are going to test our NCCF short method in both MovieLens and FilmTrust datasets. Fig. 13 shows the MovieLens comparative results for classification quality measures in both the baseline (NCF) and the proposed deep neural model (NCCF). Overall, Fig. 13 shows better precision results and worst recall results than the baseline. It is remarkable the proposed NCCF capability to maintain the precision accuracy level when a high number of recommendations is chosen. This trend is consolidated on the recall results. FilmTrust (Fig. 14) results point in the same direction: NCCF improves the baseline recall when the number of recommendations exceeds a threshold, although in this case precision is similar in both the baseline and the proposed method. The proposed method works particularly fine when the number of recommendations is not low. This is due to its original prediction schema: instead of making a prediction for each RS item, the proposed NCCF model generates a set of probabilities in the categorical output layer of the deep architecture. The result is a more equilibrated distribution of predictions along the RS set of items.

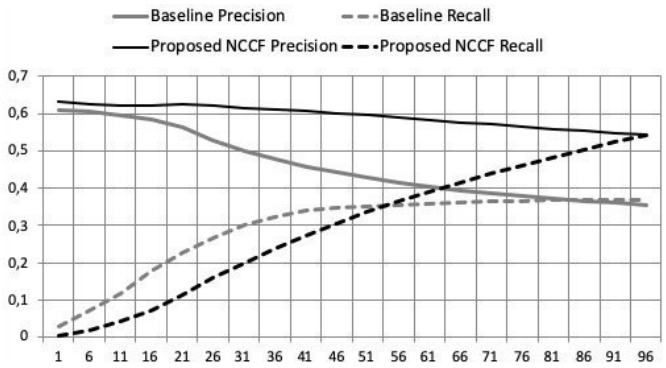


Fig. 13. MovieLens. Recommendation quality results; proposed NCCF versus the NCF baseline. NCF [33] outperforms to the state of art baselines: BPR [16], eALS [17], ItemKNN [40] and it provides similar accuracy to deepFM [34]. y-axis: Precision & recall; x-axis: Number of recommendations N.

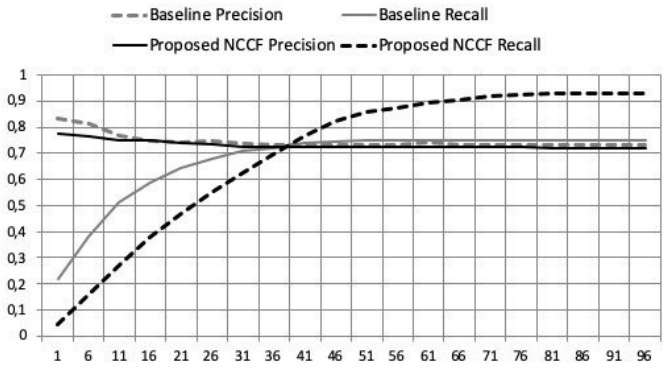


Fig. 14. FilmTrust. Recommendation quality results; proposed NCCF versus the NCF baseline. NCF [33] outperforms to the state of art baselines: BPR [16], eALS [17], ItemKNN [40] and it provides similar accuracy to deepFM [34]. y-axis: Precision & recall; x-axis: Number of recommendations N.

We can conclude by discussing that the proposed NCCF method can reach the current state of art neural collaborative filtering approaches, and it also improves recommendation results, particularly when the number of recommendations is not low. The NCCF method strengths are: 1) This is a scalable fully deep neural network approach, 2) The obtained recommendation accuracy is comparable to state of art approaches, and it improves results when the number of recommendations is not low, and 3) The size and complexity of the architecture is smaller than the autoencoder based ones.

IV. Conclusions

A scalable and original classification-based deep neural architecture has been proposed in this paper. It is based on the concept that by reducing the ratings dataset information, accuracy will not significantly change. We renounce to the numerical rating values and transform them to the binary: relevant/non-relevant vote and voted/non-voted item. This information reduction leads to a new abstraction level and it opens a new range of classification-based architectures. Two related architectures are proposed in the paper: the short one and the long one. The large architecture makes use of both relevant/non-relevant and voted/non-voted binary information, whereas the short architecture only uses the relevant/non-relevant subset of information. Experiments show that the short architecture achieves a similar performance to the large one. Even although we lose a portion of accuracy, the short architecture is chosen due to its better scalability.

The proposed NCCF model has been tested on the MovieLens and the FilmTrust datasets and their results have been compared with the state of art baseline NCF. NCF is a powerful baseline, since it outperforms KNN, MF, BPR and eALS and it provides similar accuracy to deepFM. Results show that our scalable approach provides a competitive recommendation quality. It particularly outperforms the baseline when the number of recommendations is not low. The result is consistent with the way predictions are made: to recommend a user, instead of making a prediction for each RS item, the proposed NCCF model generates a set of probabilities in the categorical output layer of the deep architecture. This is an original approach, and it obtains a more equilibrated distribution of predictions along the RS set of items.

The proposed method novelty opens a wide range of future works, among which are: a) To design an NCCF ensemble, inspired on the random forest method, where the input data that feeds each NCCF is a random subset of the RS items, b) To make use of data augmentation from the binary relevant/non-relevant information in order to enrich the set of input samples, c) To experiment the quality impact of stablishing a multilabel target on the samples, instead the proposed one label approach, and d) To incorporate a wide (memorization) layer to create a wide & deep neural network architecture.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, "Recommender systems survey," Knowledge-Based Syst., vol. 46, pp. 109–132, 2013.

[2] Xin Dong, Lei Yu, ZhonghuoWu, Yuxia Sun, Lingfeng Yuan, and Fangxi Zhang, "A hybrid collaborative filtering model with deep structure for recommender systems". In Proceedings of the AAAI.1309–1315,2017. Available: https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14676

[3] M.J. Pazzani, "A framework for collaborative, content-based and demographic filtering", Artif. Intell. Rev. 13 (5-6) pp. 393–408, 1999.

[4] T. Ebesu and Y. Fang. "Neural citation network for context-aware citation recommendation". In Proceedings of the SIGIR. ACM, Shinjuku, Tokyo, pp. 1093–1096. 2017.

[5] C. Musto, C. Greco, A. Suglia, and G. Semeraro, "Askme any rating: A content-based recommender system based on recurrent neural networks", Proceedings of the IIR, 2016.

[6] X.Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users", Proceedings of the SIGIR. ACM, pp. 185–194, 2017.

[7] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semisupervised learning: A neural approach for POI recommendation", Proceedings of the SIGKDD, pp. 1245–1254. Available: http://dl.acm.org/citation.cfm?id=3172077.3172336.

[8] B. Yi, X. Shen, Z. Zhang, J. Shu, and H. Liu, "Expanded autoencoder recommendation framework and its application in movie recommendation". Proceedings of the SKIMA, pp. 298–303, 2016.

[9] W. Chu and Y. Tsai. "A hybrid recommendation system considering visual information for predicting favorite restaurants". WWWJ (2017), pp. 1–19, 2017.

[10] R. He and J. McAuley. "Ups and downs: Modeling the visual evolution of fashion trends with oneclass collaborative filtering", Proceedings of the WWW, pp. 507–517. 2016. Available: https://doi.org/10.1145/2736277.2741667.

[11] C. Hsieh, L. Yang, H. Wei, M. Naaman, and D. Estrin, "Immersive recommendation: News and event recommendations using personal digital traces", Proceedings of the WWW, pp. 51–62. 2016. Available: https://doi.org/10.1145/2872427.2883006

[12] X. Deng, F. Huangfu, "Collaborative Variational Deep Learning for Healthcare Recommendation", IEEE Access, Vol. 7, 2019.

[13] Z. Chen, L. Li, H. Peng, Y. Liu, H. Zhu, Y. Yang, "Sparse General Non-Negative Matrix Factorization Based on Left Semi-Tensor Product", IEEE Access, Vol. 7, pp. 81599 – 81611, 2019.

[14] K. Li, X. Zhou, F. Lin, W. Zeng, G. Alterovitz, "Deep Probabilistic Matrix Factorization Framework for Online Collaborative Filtering", IEEE Access, Volume: 7, 2019.

[15] A. Hernando, F. Bobadilla, F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model", Knowledge-Based Syst. 197C (2016) pp. 188–202.

[16] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. "Bpr: Bayesian personalized ranking from implicit feedback". In UAI, pp. 452-461, 2009.

[17] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua. "Fast matrix factorization for online recommendation with implicit feedback". SIGIR, pp. 549-558, 2016.

[18] S. Zhang, L. Yao, A. Sun, and Y. Tay. "Deep Learning based Recommender System: A Survey and New Perspectives". ACM Comput. Surv.1, 1, Article 1. July 2018, pp. 35.

[19] R. Mu, "A Survey of Recommender Systems Based on Deep Learning", IEEE Access, Year: 2018 | Volume: 6, pp. 69009–69022, DOI: 10.1109/ACCESS.2018.2880197

[20] X. Wang, L. Yu, K. Ren, G. Tao, W. Zhang, Y. Yu, and J. Wang, "Dynamic attention deep model for article recommendation by learning human editors' demonstration", Proceedings of the SIGKDD, 2017, Available:

[21] http://doi.acm.org/10.1145/3097983.3098096

[22] D. Liang, M. Zhan, and D. P.W. Ellis, "Content-aware collaborative music recommendation using pre-trained neural networks", Proceedings of the ISMIR, 2015, pp. 295–301.

[23] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning". Proceedings of the MM, 2014, pp. 627–636. Available: http://doi.acm.org/10.1145/2647868.2654940

[24] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua. "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention". Proceedings of the SIGIR. 2017. ACM. Available: http://doi.acm.org/10.1145/3077136.3080797

[25] X. Chen, Y. Zhang, Q. Ai, H. Xu, J. Yan, and Z. Qin. "Personalized key frame recommendation", Proceedings of the SIGIR, 2017, pp. 315–324. Available: http://doi.acm.org/10.1145/3077136.3080776

[26] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations", Proceedings of the Recsys, 2016, pp. 241–248. Available: http://doi.acm.org/10.1145/2959100.2959167

[27] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for sesión based recommendation", Proceedings of the Recsys (RecSys'17), 2017, ACM, Como, pp. 306–310.

[28] I. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding", Proceedings of the WSDM, 2018, pp. 565–573. Available: http://doi.acm.org/10.1145/3159652.3159656

[29] T. Xuan Tuan and T. Minh Phuong, "3D convolutional networks for session-based recommendation with content features", Proceedings of the Recsys, 2017, pp. 138–146.

[30] A. Suglia, C. Greco, C. Musto, M. de Gemmis, P. Lops, and G. Semeraro, "A deep architecture for content-based recommendations exploiting recurrent neural networks", Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, 2017, ACM, Bratislava, pp. 202–211.

[31] Y. Kiam Tan, X. Xu, and Y. Liu. "Improved recurrent neural networks for session-based recommendations", Proceedings of the Recsys, 2016, pp.

17–22. Available: http://doi.acm.org/10.1145/2988450.2988452

[32] T. Alashkar, S. Jiang, S. Wang, and Y. Fu, "Examples-rules guided deep neural network for makeup recommendation", Proceedings of the AAAI, pp. 941–947. Available: https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14773

[33] S. Zhang, L. Yao, and X. Xu, "AutoSVD++: An efficient hybrid collaborative filtering model via contractive auto-encoders", ACM, 2017, pp. 957–960.

[34] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-Seng Chua, "Neural collaborative filtering", Proceedings of the WWW. 2017, pp.173–182. Available: https://doi.org/10.1145/3038912.3052569

[35] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction", Proceedings of the IJCAI, 2017, pp. 2782–2788. Available: http://dl.acm.org/citation.cfm?id=3172077.3172127

[36] G. K. Dziugaite and D. M. Roy. 2015. "Neural network matrix factorization". arXiv preprint arXiv:1511.06443 (2015).

[37] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir et al., "Wide & deep learning for recommender systems". Proceedings of the Recsys, 2016, pp. 7–10. Available: http://doi.acm.org/10.1145/2988450.2988454

[38] W. Yuan, H. Wang, B. Hu, L. Wang, Q. Wang, "Wide and Deep Model of Multi-Source Information-Aware Recommender System", IEEE Access, Year: 2018 | Volume: 6, pp. 49385-49398.

[39] S. Rendle, "Factorization machines", 2010 IEEE International Conference on Data Mining, December pp. 13-17, 2010, Sydney, Australia.

[40] J. Bobadilla, F. Serradilla, J. Bernal, "A new collaborative filtering metric that improves the behavior of recommender systems", Knowledge-Based Systems 23 (2010), pp. 520–528.

[41] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, "Item-based collaborative _ltering recommendation algorithms", WWW10, May 1-5, 2001, Hong Kong. ACM 1-58113-348-0/01/0005, pp. 285-295.

[42] H. Wang, N. Wang, D.-Y. Yeung, "Collaborative deep learning for recommender systems", Proceedings of the SIGKDD, 2015, pp.1235-1244.

[43] F. Zhang, N. Jing Yuan, D. Lian, X. Xie, W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems". Proceedings of the SIGKDD, 2015, pp. 353–362. Available: http://doi.acm.org/10.1145/2939672.2939673

[44] H. Lee, Y. Ahn, H. Lee, S. Ha, S. Lee, "Quote recommendation in dialogue using deep neural network", Proceedings of the SIGIR, 2016, pp. 957–960. Available: http://dx.doi.org/10.1145/2911451.2914734

[45] J. Golbeck, J. Hendler, "FilmTrust: movie recommendations using trust in web-based social networks", CCNC 2006, 3rd IEEE Consumer Communications and Networking Conference, 2006, DOI: 10.1109/CCNC.2006.1593032

[46] F. M. Harper and J. A. Konstan. 2015. "The MovieLens Datasets: History and Context". ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), pp. 19. Available: http://dx.doi.org/10.1145/2827872

Jesús Bobadilla

Jesús Bobadilla received the B.S. and the Ph.D. degrees in computer science from the Universidad Politécnica de Madrid and the Universidad Carlos III. Currently, he is a lecturer with the Department of Applied Intelligent Systems, Universidad Politécnica de Madrid. He is a habitual author of programming languages books working with McGraw-Hill, Ra-Ma and Alfa Omega publishers. His research interests include information retrieval, recommender systems and speech processing. He is in charge of the FilmAffinity.com research team working on the collaborative filtering kernel of the web site. He has been a researcher into the International Computer Science Institute at Berkeley University and into the Sheffield University. Head of the research group.

Fernando Ortega

Fernando Ortega was born in Madrid, Spain in 1988. He received the B.S. degree in software engineering, the M.S. degree in artificial intelligence, and the Ph.D. degree in computer science from Universidad Politécnica de Madrid, in 2010, 2011 and 2015, respectively. From 2008 to 2015, he was a Research Assistant with the Intelligent Systems for Social Learning and Virtual Environments research group. From 2015 to 2017, he was working at BigTrueData, leading machine learning projects. From 2017 to 2018, he was an Assistant Professor at U-tad: Centro Universitario de Tecnología y Arte Digital. Since 2018, he is Assistant Professor at Universidad Politécnica de Madrid. He is the author of 30 research papers in most prestigious international journals. His research interests include machine learning, data analysis and artificial intelligence. He leads several national projects to include machine learning algorithms into the society.

Abraham Gutiérrez

Abraham Gutiérrez was born in Madrid, Spain, in 1969. He received the B.S. and the Ph.D. degrees in computer science from the Universidad Politécnica de Madrid. Currently, he is a lecturer with the Department of Applied Intelligent Systems, Universidad Politécnica de Madrid. He is a habitual author of programming languages books working with McGraw-Hill, Ra-Ma and Alfa Omega publishers. His research interests include P-Systems, social networks and recommender systems. He is in charge of this group innovation issues, including the commercial projects.

Santiago Alonso

Santiago Alonso received his B.S. degree in software engineering from Universidad Autónoma de Madrid and his Ph.D. degree in computer science and artificial intelligence from Universidad Politécnica de Madrid, in 2015, where he is currently an Associate Professor, participating in master and degree subjects and doing work related with advanced databases. His main research interests include natural computing (P-Systems), and did some work on genetic algorithms. His current interests include machine learning, data analysis and artificial intelligence.

A Convolution Neural Network Engine for Sclera Recognition

M. S. Maheshan*, B. S. Harish, N. Nagadarshan

Department of Information Science and Engineering, JSS Science and Technology University, Mysuru (India)

Received 28 November 2018 | Accepted 20 March 2019 | Published 22 March 2019

Abstract

The world is shifting to the digital era in an enormous pace. This rise in the digital technology has created plenty of applications in the digital space, which demands a secured environment for transacting and authenticating the genuineness of end users. Biometric systems and its applications has seen great potentials in its usability in the tech industries. Among various biometric traits, sclera trait is attracting researchers from experimenting and exploring its characteristics for recognition systems. This paper, which is first of its kind, explores the power of Convolution Neural Network (CNN) for sclera recognition by developing a neural model that trains its neural engine for a recognition system. To do so, the proposed work uses the standard benchmark dataset called Sclera Segmentation and Recognition Benchmarking Competition (SSRBC 2015) dataset, which comprises of 734 images which are captured at different viewing angles from 30 different classes. The proposed methodology results showcases the potential of neural learning towards sclera recognition system.

Keywords

Biometrics, Convolution Neural Network, Deep Learning, Sclera Recognition.

DOI: 10.9781/ijimai.2019.03.006

I. Introduction

THERE is an exponential growth in digital technology associated with the increased application in commerce and telecommunication throughout the world which demands a strong authentication system. Consequently, there is a constant search for novel and secured system for authentication. In this regard biometric system of identification and authentication is getting more popular due to its reliability and uniqueness. Biometric authentication to individuals premise that everyone is unique with their physical or behavior traits which includes face [1], [2], iris [3], [4], fingerprints [5], finger vein patterns [6], palm prints [7], etc. Sclera is one such trait which is gaining lot of attention in research community. Sclera is the white outer layer of the eyeball as shown in Fig. 1

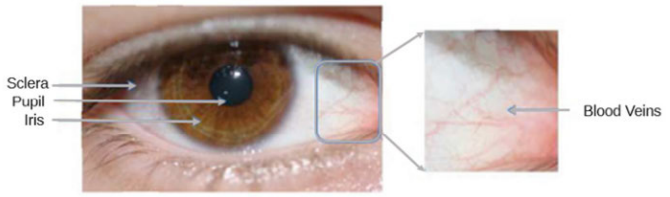


Fig. 1. Structure of the eye and its sclera region.

Sclera trait has emerged as most promising to complement other traditional traits. This is because sclera is a highly-protected portion of the eye which is very difficult to forge [8].The vessel pattern of the sclera is observed to be never the same for any two individuals, including the identical twins, and the pattern even differs with the left eye and the right eye in each individual. Authentication of a person by the vessel pattern of the sclera is possible due to its high degree of randomness and uniqueness. Hence, it is necessary to develop an efficient and robust scleral pattern recognition system for person authentication. This can be tackled by applying deep learning techniques such as Convolution Neural Network (CNN), Recurrent Neural Network (RNN), Recursive Neural Networks, etc. Convolution Neural Network (CNN) and its approaches have become most talked topic in the field of deep learning. The neural learning model solutions have been most successful than ever before by making the computer learn the model and yield effective results especially for image segmentation [29] and detection [30]. This carves a breakthrough by providing state of the art results to many applications such as face recognition [9], object detection [10], driver monitoring system [11], etc.

The rest of the paper is structured as follows. Section 2 reviews the background work, section 3 presents the proposed model engine for recognition, followed by discussion on dataset and result analysis in section 4 and later the paper concludes with future scope in section 5.

II. Background

Biometric trait has been integrated to a wide spectrum of applications and services. Sclera trait is emerging as it has more promising trait to complement the traditional traits, this is because of its pattern uniqueness and randomness. Hence this demands to develop an efficient and robust scleral pattern recognition system for person authentication. Over the past decade, researchers have proposed different algorithms for segmentation and recognition. However, there are many unsolved challenges which paves way to open research in this area. The premise of sclera authentication system is segmentation

* Corresponding author.
E-mail addresses: maheshan@jssstuniv.in

and recognition. Typically when the image is taken as input, it is preprocessed to get rid of occluded portions of the eye and segmentation process extracts the relevant data objects from the image [12]. Further, the selective region of interest is subjected to feature extraction and is thus matched against the knowledge base for recognition. Fig. 2 depicts the overall process of sclera segmentation and its recognition.

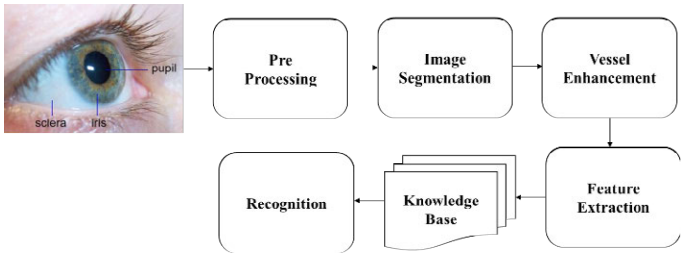


Fig. 2. Typical process of sclera segmentation and recognition process.

A lot of importance has been given to segmentation stage, where in a lot of studies such as TASOM (Time Adaptive Self-Organizing Map)-based active contour [13], [14] method proposed to get the inner boundary of the sclera. For non-ideal conditions, Crihalmeanu et al., [15] used K-means clustering to segment the sclera by matching conjunctival vasculature of an eye. For the colour images [16], the HSV model was used to segment the eye images automatically. The prominence of the blood veins in the sclera region is not striking, for which image enhancement plays a vital role in recognition system. In [17], the authors used Histogram Equalization (HE) and Contrast Limited Adaptive Histogram Equalization (CLAHE) for enhancing the images, and later combined the methods with K-means and FCM for which it was found that HE with K-means had better precision rate when compared to CLAHE with K-means and FCM. An adaptive Histogram Equalization was applied to the green layer of the colour image in order to get enhanced sclera vein pattern in [18] – [21]. The features of interest for effective sclera recognition build an authentic mathematical model for authentication and identification purposes. In [18], the Discrete Cosine Transform (DCT) and wavelets were used for feature representation. A Local Binary Pattern (LBP) feature was used for extracting texture based feature and the effect of feature fusing in sclera biometrics was studied in [22]. The template matching algorithm was proposed for classification in [16]. A template based matching was introduced using hamming distance for classification in [23], [24]. For recognition, a benchmarking competition was organized to record the recent advancements in recognition techniques where the winning team achieved 72.56% accuracy in eye recognition [25]. From various literature, it is observed that many challenges are being addressed by various researchers to improve the accuracy of the recognition system. Convolution Neural Network (CNN) in recent times has set benchmarking results on various computer vision tasks and applications including the biometrics [26], [27]. There are various advanced deep learning architectures like GoogleNet, AlexNet, ResNet, etc., applied to various classification problems. Authors in [32] present a simple unsupervised convolutional deep learning network called PCANet for image classification in cascaded way to learn multi stage filter banks which are further processed to binary hashing and block histograms for indexing and pooling. For hyperspectral image classification, a new model R-VCANet was introduced in [33] which reveals better results even though if there exists less training samples. This method used the inherent properties of HSI data like spatial information and spectral characteristics in order to improve the feature expression of the network. Authors in [34] present a CNN model to embed visual words in images. This method treated images as textual document, built visual words and embedded them to capture the spatial context surrounding them. This method performed better than the original images and showed how to bridge between embedded text data and how to adapt for visual data in a wide range of generic image and video applications. All the above mentioned works have attempted using deep learning networks for image classification. However for sclera recognition, there are no findings of usage of deep learning in general and convolution neural nets in specific. This brings us to explore the neural model to next level by proposing Convolutional Neural Network Sclera Recognition Engine (CNNSRE) for sclera recognition system. The proposed method is a simplest form of deep learning model. Hence there are lot of potential avenues that can be explored using neural model for building sclera recognition system.

III. Proposed Model

The proposed model is first of its kind to propose CNN based approach for a sclera recognition system. Although it appears to be at the initial level of deep learning architecture, unlike the traditional approaches, this gives the opportunity to testify the power of deep learning towards sclera biometric system. As there are no attempts made towards recognition process of sclera biometric using deep learning models, we take the initiative to explore and learn how the proposed model works for sclera recognition system. The implementation of this model is done using keras [31] open source package in python language. Fig. 3 presents the proposed Convolutional Neural Network Sclera Recognition Engine. The Convolution Neural Network architecture consists of 4 main convolutional units, each having a convolutional layer, out of these some are followed by a Max Pooling layer and 1 fully connected unit. The CNNSRE contains 5 main units, namely:

A. First Convolutional Unit

B. Second Convolutional Unit

C. Third Convolutional Unit

D. Fourth Convolutional Unit

E. Fully Connected Unit (FCL)

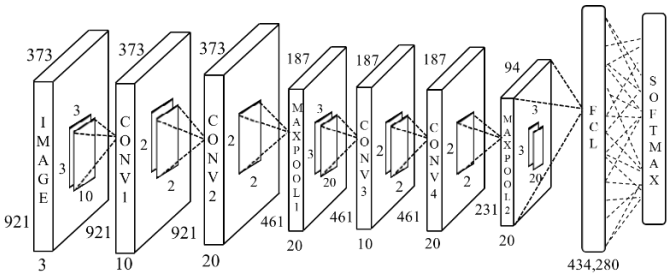


Fig. 3. Convolutional Neural Network Sclera Recognition Engine (CNNSRE).

A. First Convolutional Unit

The CNN model requires all the sample images to be of the same size. In this regard we rescaled the images to 373 X 921 X 3, and this is done by considering the minimum height of all the images which is 373 and the minimum width of all the images which is 921. The first convolutional unit consist of 10 filters of size 3 X 3 X 3. The 10 filters is due to the limitation in availability of computational resources. The depth of the 10 filters is 3 because of the input image, which is an RGB and the depth of an RGB image is 3. To retrieve the same dimension of the images in the next layer, we zero-pad the input image before convolving the filters on the image of size 373 X 921 X 3. Another parameter is striding which is kept as 1 because there could be chances of missing important features in the input image. The output after convolving the filters on the image will be of size 373 X 921 X 10. The output's height and width are same as input because we zero-padded the image and depth as 10 as we are using 10 filters. Equation

(1) shows the dot product that happens when we convolve the filters on the image.

$$f(x) = Wx + b \tag{1}$$

Where, W is the weight matrix of the filters

x is the pixel value of the image

b is the bias

After convolving the filters, the output of dimension 373 X 921 X 10 is fed to the Rectified Linear Unit (ReLU) activation unit. Choosing ReLU as an activation function over sigmoid and Tanh is because sigmoid has a problem called as vanishing gradient problem. The gradient could approach to zero and no weight update could take place resulting in no learning. The other problem of the sigmoid is that it has an exponential term which increases the complexity. Tanh activation function solves the zero centered problem but not the important vanishing gradient problem. However, ReLU solves above problems and hence this is being used in our proposed model.

$$R(z) = \max(0, z) \tag{2}$$

The range of the ReLU activation function lies between [0, ∞] and the equation for the same is given in (2). This will give zero for negative numbers and for the rest, the output will be same as input. After employing the ReLU activation function the image is now fed to second convolution unit.

B. Second Convolutional Unit

The second convolutional unit is similar to the first convolutional unit. The only change is in the number of filters used and an addition of max pooling layer. As earlier, at first we convolve the filters over the output of the previous convolutional layer. The output image from the first convolution layer is of dimension 373 X 921 X 10. In this unit, we are increasing the filter count to 20. The depth of each of the 20 filters in this layer is set to 10, because the depth of the output of the previous layer was 10. In this unit the stride parameter for convolutional filters is kept as 1. The input is zero padded so as to get the output which is of dimension 373 X 921 for each filter. The output thus after convolving the image will become 373 X 921 X 20 as there are 20 filters. The input image is then fed to the ReLU activation unit as done in the previous layer. Further getting the output from the ReLU activation unit, the output is inputted to the 2 X 2 max pooling layer with stride 2. This is because the image would get overlapped with the previous image if the stride is kept to 1. Therefore, the output of the second convolutional unit becomes 187 X 461 X 20.

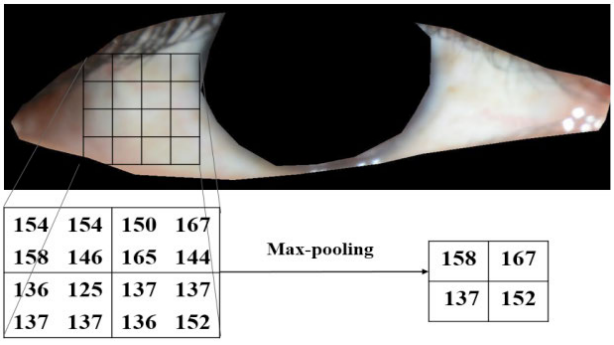


Fig. 4. 2 X 2 max pooling operations.

This change in dimension is because of max pooling layer as shown in Fig. 4. This figure depicts a small max pooling operation for the 2 X 2 grid. In the 2 X 2 max pooling operations, the whole image is divided into 2 X 2 grids because 1 X 1 keeps the same dimension and 3 X 3 could down sample the image in large scale. From each grid maximum pixel unit is taken i.e., 158 from first grid, 167 from the second grid, 137 from the third grid, and 152 from the last grid forms a new image after 2 X 2 max pooling.

C. Third Convolutional Unit

The third convolutional unit is similar to the first convolutional unit. As earlier, at first, we convolve the filters over the output of the previous convolutional layer. The output image from the second convolution unit is of dimension 187 X 461 X 20. In this unit we are considering the filter count as 10. The depth of each of the 10 filters in this layer is set to 20, because the depth of the output of the previous layer was 20. Thus in this unit the stride parameter for convolutional filters is kept as 1. The input is zero padded so as to get the output which is of dimension 187 X 461 for each filter. The output thus after convolving the image will become 187 X 461 X 10 as there are 20 filters. The input image is then fed to the ReLU activation unit as done in the previous layer. After getting the output from the ReLU activation unit, the output with dimensions 187 X 461 X 10 is fed to the fourth convolutional unit.

D. Fourth Convolutional Unit

The fourth convolutional unit is similar to the second convolutional unit. At first, we convolve the filters over the output of the previous convolutional layer. The output image from the second convolution unit is of dimension 187 X 461 X 10. In this unit we are considering the filter count as 20. The depth of each of the 20 filters in this layer is set to 10, because the depth of the output of the previous layer was 20. In this layer, the stride parameter for convolutional filters is kept as 1. The input is zero padded so as to get the output which is of dimension 187 X 461 for each filter. The output thus after convolving the image will become 187 X 461 X 20 as there are 20 filters. The input image is then fed to the ReLU activation unit as done in the previous layer. After getting the output from the ReLU activation unit, it is inputted to the 2 X 2 max pooling layer with stride 2. If stride was not 2, then the image would get overlapped with the previous image (if the stride was kept to 1). Therefore, the output of the second convolutional unit becomes 94 X 231 X 20. After flattening the 94 X 231 X 20 output to 434,280 dimensional vector, it is fed to the input of the fully connected unit.

E. Fully Connected Unit

This is the fifth unit of the proposed model. Here the output of the previous layer is taken as input to the Fully Connected Layer (FCL) of 434,280 neurons. This is because of converting the matrix of dimension 94 X 231 X 20 into a vector. The output of fully connected layer is fed to softmax layer which outputs 30 scores because of 30 classes i.e., 30 different individual's data. The purpose of using the softmax function as the loss function, is to convert the linear inputs to probabilities i.e., to range the scores between 0 and 1 which is easier to compute.

The softmax function is given in (3):

$$softmax(y_i) = \frac{\exp(y_i)}{\sum_j \exp(y_j)} \tag{3}$$

Where $exp(y_i)$ is the exponential function and i denotes the i^{th} class. The denominator $exp(y_j)$ is the summation of exponential scores of all the classes. After calculating the loss, the model back propagates the errors and update the weights. The number of epochs was set to 100 to train the CNNSRE to achieve the highest accuracy of 87.65%. When the epochs was set above 100, the model met its threshold due to the computational power, and when the epochs was set below 100, the results was degraded. Table I presents the characteristic of CNNSRE which describes the sclera recognition engine with its number of layers, type of layers, number of filters, dimension of feature maps, kernel size, striding and padding.

IV. Dataset and Result Analysis

To validate the strength of the proposed method, experimentation is carried on Sclera Segmentation and Recognition Benchmarking Competition (SSRBC2015) dataset [28]. This dataset contains 30 individuals' eye images of different cases such as blurred, blinking and closed eye. For every individual eye, multi-angle images are captured in different views like center, left, right and up. Thus, dataset in total consists of 734 segmented eye images from all 30 classes. The computational complexity is cubic for training and linear for testing. The determined complexities for training time is shown in (4) and testing time in (5).

$$O(n^2.m) \tag{4}$$

Where, n is number of train images

m is number of epochs

$$O(n) \tag{5}$$

Where, n is number of test images.

Table II presents the precision, recall, False Rejection Rate (FRR), Genuine Acceptance Rate (GAR), and False Acceptance Rate (FAR) for individual classes of the dataset. The model was run for 100 epochs because of computational constraints for which the result achieved a maximum of 87.65% accuracy rate and an average of 83.76% accuracy rate and a least of 81.17% accuracy rate. The variation of accuracy is due to the random initialization of weights in the model. The model was again tested against the scores of precision, recall, FRR, GAR, and FAR. It is been very interesting to notice the precision score of 0.86, for the sclera recognition. GAR scored 0.85 on an average, and FRR scored 0.13 and FAR scored 0.015. This shows that deep learning models can perform better and show less error rate for sclera recognition. Table III presents the results of the proposed model over the existing models presented by different authors in SSRBC 2016 competition. The table presents only two works, this is because of very limited work towards recognition. Among the two works, the participating team Sl. No.1 from the table III proposed KNN based sclera recognition system by two simple steps of feature extraction and matching. For feature extraction they used Histogram of Oriented Gradient (HOG) descriptor to extract the features and match them against the training tuples by comparing a given test tuple with training tuples which are similar to it. The other team Sl. No.2 uses multiclass feature-based classifier for recognition. They employed a 2D Gabor filter, which down sampled the feature vector to size 720, and KNN for classification for which resulted in a hike of 1.56% when compared to

TABLE II. Individual Class Results of SCLERA Recognition Model

Class	Precision	Recall	FRR	GAR	FAR
1	0.8	0.36	0.53	0.36	0.002
2	0.4	0.3	1	0.5	0.3
3	1	0.66	0.33	0.66	0
4	0.75	0.5	0.5	0.5	0.005
5	0.79	1	0	1	0.008
6	1	1	0	1	0
7	0.88	1	0	1	0.002
8	0.76	0.89	0.12	0.87	0.02
9	0.61	1	0.67	0.12	0.029
10	0.5	0.87	0.38	0.61	0.041
11	0.5	0.12	0.61	0.38	0.005
12	0.6	0.61	0	1	0.032
13	0.6	0.38	0	1	0.02
14	0.72	1	0	1	0.008
15	1	1	0	1	0
16	1	1	0	1	0
17	1	1	0	1	0
18	1	1	0	1	0
19	1	1	0	1	0
20	1	1	0	1	0
21	1	1	0	1	0
22	1	1	0	1	0
23	1	1	0	1	0
24	0.91	1	0	1	0.002
25	1	1	0	1	0
26	1	1	0	1	0
27	1	1	0	1	0
28	1	1	0	1	0
29	1	1	0	1	0
30	1	1	0	1	0
Average	0.860	0.856	0.138	0.866	0.015

TABLE I. Characteristics of CNNSRE

	Type of Layer	No. of filters	Feature Map Size (Height X Width X Depth)	Kernel size	No. of stride	No. of padding
	Image Input Layer	–	373 X 921 X 3	–	–	–
1st Convolutional Unit	Conv – 1 (1st Convolutional Layer)	10	373 X 921 X 10	3 X 3	1 X 1	1 X 1
2nd Convolutional Unit	Conv - 2 (2nd Convolutional Layer)	20	187 X 461 X 20	3 X 3	1 X 1	1 X 1
	Max-Pooling – 1 (1st pooling layer)	1	187 X 461 X 10	2 X 2	2 X 2	0 X 0
3rd Convolutional Unit	Conv - 3 (3rd Convolutional Layer)	10	187 X 461 X 10	3 X 3	1 X 1	1 X 1
4th Convolutional Unit	Conv - 4 (4th Convolutional Layer)	20	187 X 461 X 20	3 X 3	1 X 1	1 X 1
	Max-Pooling – 2 (2nd pooling layer)	1	94 X 231 X 20	2 X 2	2 X 2	0 X 0
Fully Connected Unit	Fully Connected Layer Softmax Output	–	434,280 X 1 30 X 1 30 X 1	–	–	–

the first team. The proposed model using CNN exploits the knowledge of self learning by training its network architecture i.e., at every layer the images are convolved and output result is fed to the next layers. By doing so the model learns to correct its errors by updating the weights which happens by back propagation. This repetition helps the model to train effectively by correcting the errors and yielding better results. This kind of approach is possible through deep learning architectures. In the traditional approach like KNN, the testing time is more complex rather than training. However using CNN in biometrics the testing should be faster than training for ease of applications.

TABLE III.
Overall Result of Proposed Model vs. Other Competition Models

Sl No.	Proposed model vs. Existing models	Accuracy
1	Aruna Kumar S V (SJCE, Mysuru, Karnataka, India) / segmentation and recognition(SSRBC 2016)	80.55
2	Chandranath Adak (Griffith University,Australia) & BhageshSeraogi (ISI, Kolkata, India) / recognition (SSRBC 2016)	82.11
3	**Proposed Model**	**87.65**

V. Conclusion

This work is first of its kind in applying convolution neural network towards sclera recognition which showcases the potential of neural learning to biometric applications. It is found that there are good amount of works done towards segmentation with competition results. However there are limited contributions in sclera recognition system. Hence, in this work we presented CNN model with four convolution layers and with the limited computational resource, the result table shows that the maximum accuracy achieved is 87.65%. The CNN model could be trained more by providing increased learning data, with more classes and the results could be improved by adding more layers in to the model and randomizing the weights with more computational power. The sclera recognition with deep learning concepts being successful in this work shall be able to open plethora of opportunities to researchers to work on recognition model in near future. Further the results can be enhanced for more accuracy by exploring, training and developing complex neural models for both segmentation and recognition. In future, we also intend to exploit the concept of CNN with neuromorphic models with CNN networks. This hybridization will be responsible to capture more realistic sense of data which in-turn improvises the results to the best extent.

References

[1] G. Larrain, T.; Bernhard, J.S.; Mery, D.; Bowyer, K.W. Face recognition using sparse fingerprint classification algorithm. IEEE Trans. Inf. Forensics Secur., vol. 12, pp. 1646-1657, 2017.

[2] Bonnen, K.; Klare, B.F.; Jain, A.K. Component-based representation in automated face recognition. IEEE Trans. Inf. Forensics Secur. vol. 8, 239–253, 2013.

[3] Daugman, J. How iris recognition works. IEEE Trans. Circuits Syst. Video Technol., vol. 14, pp. 21-30, 2004.

[4] Viriri, S.; Tapamo, J.R. Integrating iris and signature traits for personal authentication using user-specific weighting. Sensors vol. 12, 4324–4338, 2012.

[5] Jain, A.K.; Arora, S.S.; Cao, K.; Best-Rowden, L.; Bhatnagar, A. Fingerprint recognition of young children. IEEE Trans. Inf. Forensic Secur, vol. 12, pp. 1501-1514, 2017.

[6] Hong, H.G.; Lee, M.B.; Park, K.R. Convolutional neural network-based finger-vein recognition using NIR image sensors. Sensors 2017, vol. 17, 1297.

[7] Yaxin, Z.; Huanhuan, L.; Xuefei, G.; Lili, L. Palmprint recognition based on multi- feature integration. In Proceedings of the IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, Xi'an, China, 3–5 October 2016; pp. 992–995.

[8] Das, A., Pal, U., Ballester, M.A.F., Blumenstein, M., 2014. A new efficient and adaptive sclera recognition system, in: Computational Intelligence in Biometrics and Identity Management (CIBIM), 2014 IEEE Symposium on, IEEE. pp. 1–8.

[9] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.

[10] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in Advances in neural information processing systems, 2013, pp. 2553–2561.

[11] J. Lemley, S. Bazrafkan, and P. Corcoran, "Transfer Learning of Temporal Information for Driver Action Classification," in The 28th Modern Artificial Intelligence and Cognitive Science Conference (MAICS), 2017.

[12] Kuruvilla, J., Sukumaran, D., Sankar, A., Joy, S.P., 2016. A review on image processing and image segmentation, in: Data Mining and Advanced Computing (SAPIENCE), International Conference on, IEEE. pp. 198–203.

[13] H. Shah-Hosseini, R. Safabakhsh, TASOM: a new time adaptive self organizing map, IEEE Trans. Syst. Man Cybern. Part B, vol. 33, no. 2, pp. 271-282, 2003.

[14] H. Shah-Hosseini, R. Safabakhsh, A TASOM-based algorithm for active contour modelling,Pattern Recognition Letter, vol. 24, no. 9, pp. 1361-1373, 2003.

[15] S. Crihalmeanu , A. Ross, and R. Derakhshani, Enhancement and Registration Schemes for Matching Conjunctival Vasculature, Appeared in Procceding of the 3rd IAPR/IEEE International Conference on Biometrics, pp.1247-1256, 2009

[16] Z. Zhou, Y. Du, N. L. Thomas, and E. J. Delp, Multi-angle Sclera Recognition System, IEEE Workshop on Computational Intelligence in Biometrics and Identity Management: pp. 103-108, 2011.

[17] Maheshan, M. S., B. S. Harish, and N. Nagadarshan. "On the use of Image Enhancement Technique towards Robust Sclera Segmentation." Procedia Computer Science, vol. 143, pp. 466-473, 2018.

[18] R. Derakhshani and A. Ross, A texture-based neural network classifier for biometric identification using ocular surface vasculature, Proc. of International Joint Conference on Neural Networks, pp. 2982-2987, 2007.

[19] S. P. Tankasala, P. Doynov, R. R. Derakhshani, A. Ross and S. Crihalmeanu, Biometric Recognition of Conjunctival Vasculature using GLCM Features, International Conference on Image Information Processing, pp. 1-6, 2011.

[20] V. Gottemukkula, S. K. Saripalle, S. P. Tankasala, R. Derakhshani, R. Pasula and A. Ross, Fusing Iris and Conjunctival Vasculature: Ocular Biometrics in the Visible Spectrum, IEEE Conference on Technologies for Homeland Security, pp. 150-155, 2012.

[21] S. Crihalmeanu , A. Ross, and R. Derakhshani, Enhancement and Registration Schemes for Matching Conjunctival Vasculature, Appeared in Procceding of the 3rd IAPR/IEEE International Conference on Biometrics, pp.1247-1256, 2009

[22] K. Oh and K. Toh, Extracting Sclera Features for Cancelable Identity Verification, 5th IAPR International Conference on Biometric, pp. 245-250, 2012.

[23] Z. Zhou, Y. Du, N. L. Thomas, and E. J. Delp, A New Human Identification Method: Sclera Recognition, IEEE transaction on System, Man And Cybernatics PART A: System And Human, vol. 42, no. 3, pp - 571-583, 2012.

[24] Z. Zhou, Y. Du, N. L. Thomas, and E. J. Delp, Multimodal eye recognition. Proc. of the International Society for Optical Engineering, vol.7708, no. 770806, pp. 1-10, 2010.

[25] Das, Abhijit, Umapada Pal, Miguel A. Ferrer, Michael Blumenstein, DejanŠtepec, Peter Rot, ŽigaEmeršič et al. "SSERBC 2017: Sclera segmentation and eye recognition benchmarking competition." In Biometrics (IJCB), 2017 IEEE International Joint Conference on, pp. 742-747, IEEE, 2017.

[26] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the Gap

to Human-levelPerformance in Face Verification. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1701–1708, 2014.
[27] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. In British Machine Vision Conference BMVC, vol. 1, pp. 6, 2015.
[28] Das, A., Pal, U., Ferrer, M.A., Blumenstein, M., 2016. Ssrbc 2016: sclera segmentation and recognition benchmarking competition, in: Biometrics (ICB), 2016 International Conference on, IEEE. pp. 1–6.
[29] J. Long, E. Schelhamar, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 3431–3440.
[30] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in Proc. Adv. Neural Inf. Process. Syst. Conf., 2013, pp. 2553–2561.
[31] Keras: The Python Deep Learning library - https://keras.io/
[32] Chan, Tsung-Han, et al. "PCANet: A simple deep learning baseline for image classification?" IEEE transactions on image processing, vol. 24, no. 12, pp. 5017-5032, 2015.
[33] Pan, Bin, Zhenwei Shi, and Xia Xu. "R-VCANet: A new deep-learning-based hyperspectral image classification method." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 5, pp. 1975-1986, 2017.
[34] Saleh, Adel, et al. "Deep visual embedding for image classification." 2018 International Conference on Innovative Trends in Computer Engineering (ITCE). IEEE, 2018.

M S Maheshan

M S Maheshan is currently working as Assistant Professor in the Department of Information Science & Engineering, JSS Science and Technology University, Mysuru, India. He obtained his M.Tech in Information Technology from SRM University, Chennai and is pursuing his Ph.D. in JSS Science and Technology University. His area of interests includes Biometrics, Soft Computing, and Machine Learning. He has certifications from British Council for English Executive course at Upper-Intermediate level, Microsoft Technology Associate on Security Fundamentals. He is Associate Member of Institute of Engineers (AM1716339).

B S Harish

B S Harish obtained his B.E in Electronics and Communication (2002), M.Tech in Networking and Internet Engineering (2004) from Visvesvaraya Technological University. He completed his Ph.D. in Computer Science (2011); thesis entitled "Classification of Large Text Data" from University of Mysuru. He is presently working as an Associate Professor in the Department of Information Science & Engineering, JSS Science and Technology University, Mysuru. He is also serving and served as a reviewer for National, International Conferences and Journals. He has published papers in more than 64 International reputed peer reviewed journals and conferences proceedings. He also served as a secretary, CSI Mysore chapter. He is a Member of IEEE (93068688), Life Member of CSI, Life Member of Institute of Engineers and Life Member of ISTE. His area of interest includes Machine Learning, Text Mining and Computational Intelligence.

N Nagadarshan

N Nagadarshanis currently studying his final year of engineering in the Department of Information Science & Engineering, JSS Science and Technology University, Mysuru. His area of interests are Deep Learning, Biometrics and also very much passionate about Spiking Neural Networks. He also interned at Indian Statistical Institute, Kolkata. Currently interning at Sabre Corporation and also served as treasurer for Computer Society of India, SJCE, Mysuru.

Deep Neural Networks for Speech Enhancement in Complex-Noisy Environments

Nasir Saleem*, Muhammad Irfan Khattak

Department of Electrical Engineering, Gomal University, D.I.Khan (Pakistan)
Department of Electrical Engineering, University of Engineering & Technology, Peshawar, Kohat Campus (Pakistan)

Received 9 February 2019 | Accepted 31 May 2019 | Published 18 June 2019

Abstract

In this paper, we considered the problem of the speech enhancement similar to the real-world environments where several complex noise sources simultaneously degrade the quality and intelligibility of a target speech. The existing literature on the speech enhancement principally focuses on the presence of one noise source in mixture signals. However, in real-world situations, we generally face and attempt to improve the quality and intelligibility of speech where various complex stationary and nonstationary noise sources are simultaneously mixed with the target speech. Here, we have used deep learning for speech enhancement in complex-noisy environments and used ideal binary mask (IBM) as a binary classification function by using deep neural networks (DNNs). IBM is used as a target function during training and the trained DNNs are used to estimate IBM during enhancement stage. The estimated target function is then applied to the complex-noisy mixtures to obtain the target speech. The mean square error (MSE) is used as an objective cost function at various epochs. The experimental results at different input signal-to-noise ratio (SNR) showed that DNN-based complex-noisy speech enhancement outperformed the competing methods in terms of speech quality by using perceptual evaluation of speech quality (PESQ), segmental signal-to-noise ratio (SNRSeg), log-likelihood ratio (LLR), weighted spectral slope (WSS). Moreover, short-time objective intelligibility (STOI) reinforced the better speech intelligibility.

Keywords

Deep Neural Networks, Intelligibility, Deep Learning, Speech Enhancement, Time-Frequency Masking, Ideal Binary Mask.

DOI: 10.9781/ijimai.2019.06.001

I. Introduction

SPEECH enhancement is a vital research problem in many audio and speech signal processing applications. The aim of the speech enhancement is to improve the quality and intelligibility of a noisy speech signal. In applications such as hearing aids, automatic speech recognition (ASR), mobile communication etc., speech enhancement has been an active research area and countless approaches have been proposed in the recent past to solve this problem [1]-[7]. One of the simplest methods to eliminate the additive background noise was spectral subtraction proposed by Boll [8]. The wiener filtering [9] based method was proposed to estimate the noise in means square error (MSE) manner. Another important method is MMSE [10], which performs nonlinear estimation of the short-time spectral amplitude (STSA) of the speech signals. An excellent adaptation of the MMSE estimation, acknowledged as Log-MMSE attempts to minimize the MSE in the log-spectral domain [11]. Additional approaches include the signal-subspace [12], sparse coding [13], and empirical mode decomposition (EMD) [14] based methods, which are frequently used to perform the task of speech enhancement.

Recently deep neural networks (DNNs) based deep learning architectures have been found to be exceptionally successful in the automatic speech recognition (ASR) [15]-[16]. This achievement of DNNs in ASR directed to investigate the DNNs for noise elimination for speech enhancement [3], [17]-[19]. Fig. 1 shows the DNN based speech enhancement framework. The key idea behind using a deep neural network for speech enhancement is that, the degradation of the speech by noise signal is a difficult process and a complex nonlinear architecture like deep neural network is suitable to model it. A very few in-depth studies based on DNNs for speech enhancement are available in the literature; however, DNNs have shown remarkable outcomes and outperformed many classical speech enhancement methods. A general feature of such studies [18] [20] is evaluated in matching noise conditions. Matching noise conditions implies that the testing noise source is similar to training noise source. Mismatched noise a condition means to the situations when a DNN model has not been aware of testing noise sources during training. Xu in [18] provided a prominent study related to speech enhancement in mismatch conditions using DNNs. During this study, DNN was trained based on the variety of noise sources and showed that large improvements are achievable in mismatched conditions by exposing DNNs to a large number of noise sources. Mismatched noise conditions are relatively difficult scenarios compared to matched conditions. In real-world environments, we expect the DNN not to only execute well on large noise sources but also on nonstationary noises. Generally, speech signals are degraded by multiple noise sources in the real world situations and therefore elimination of single noise source in previous works is limiting. In environments around us, multiple noise sources simultaneously mix with the target speech and this multiple noise types situations are obviously much difficult to eliminate/suppress. To examine the speech

* Corresponding author.
E-mail address: nasirsaleem@gu.edu.pk

enhancement in such complex nonstationary situations, we suggest moving to an environment explicit prototype.

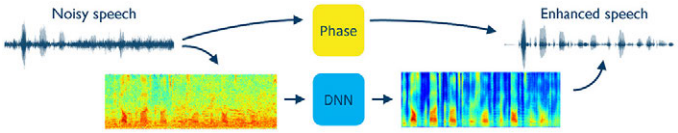


Fig. 1. DNN based Speech Enhancement.

In this paper we focus on a situation where complex noises are combined, for example, people talking in the street while vehicles are moving and the construction works are in progress. We synthesized complex noisy stimuli by adding multi-talker babble, airport and street noise simultaneously with clean utterances at different input SNRs. We have considered the deep learning based approach for speech enhancement in complex-noisy environments and an ideal binary mask (IBM) is considered as a binary classification function by using deep neural network (DNN). IBM is used as a target function during training and testing; the trained DNNs are used to estimate the IBM. The mean square error (MSE) is used as an objective cost function. The estimated target function is then applied to multi-noise mixtures to obtain the target speech.

The remaining paper is organized as: section II describes the basic problem and training DNNs for speech enhancement in complex-noisy environments, section III provides an explanation of the experiments and results and finally we concluded in section IV.

II. DNN Based Speech Enhancement

Our objective is to enhance a noisy speech in the complex-noisy conditions; where a number of possible different noise sources simultaneously degrades the quality of target speech utterances. The complex-noisy environments contain both stationary as well as nonstationary complex noise sources of completely different acoustic characteristics and are very close to real-world environments. Fig. 2 shows the time-domain waveforms and power spectral densities of various noise sources. Speech degradation under such conditions is a difficult and complex process compared to the single noise source, consequently enhancement of noisy speech becomes a more difficult task. Deep Neural Networks have high non-linear modeling capabilities and are presented in this paper for speech enhancement in complex multi-noise conditions. Prior to the actual DNN depiction, it is imperative to specify target function for DNN processing. Ideal ratio mask (IRM), ideal binary mask (IBM), short-time Fourier transform (STFT) magnitude and its mask, Mel-frequency spectrum and log-power spectra are potential target functions. We have selected IBM as target function [21]. During training, DNNs are trained and features are extracted from the noisy as well as clean speech utterances. A combined version of MFCC and RASTA-PLP features [22] are used in this paper. The extracted features are coupled with delta features to obtain Δ+DNN models. The time-frequency (T-F) representation utilized to create IBM which used a gammatone filter bank having 64 linearly spaced filters on a MEL frequency scale 50 Hz to 8 kHz and a bandwidth is equal to one Equivalent Rectangular Bandwidth (ERB) [23]. The output of the filter bank is divided into 20 ms frames with 10 ms overlap and with sampling frequency of 16 kHz. Let the noisy speech given as:

$$y(t) = s(t) + d(t) \tag{1}$$

Where $s(t)$ and $d(t)$ denote the clean speech and noise signals, respectively. The frequency-domain depiction of $y(t)$ is obtained as:

$$Y(k,\omega) = S(k,\omega) + D(k,\omega) \tag{2}$$

Where, ω and k denote frequency bin and time frame. During enhancement/testing, trained DNN is supplied with the features of the noisy speech to predict the coefficients of time-frequency mask. We have computed the coefficients of IBM, given as:

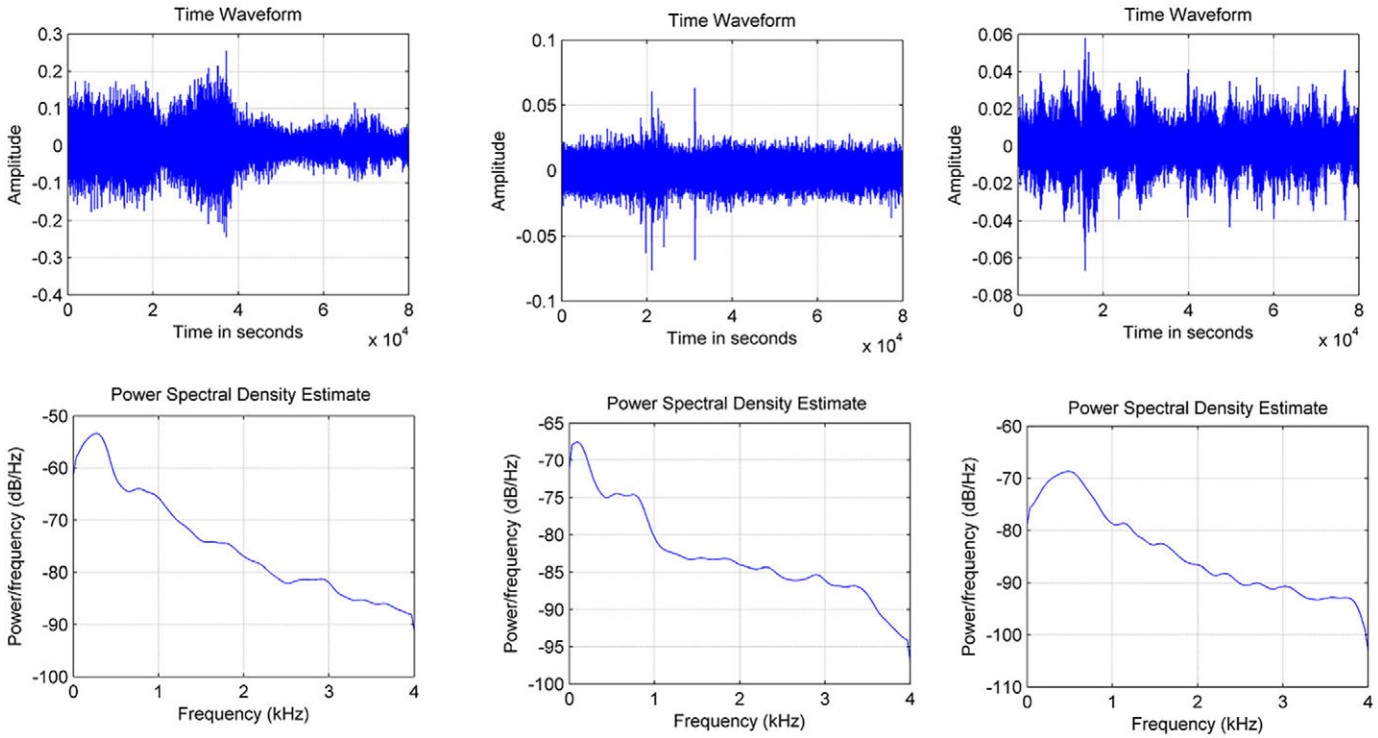


Fig. 2. Time-domain waveforms and Power Spectral Densities of Noise sources:Street, Exhibition hall and 32-multitakler babble noise (From left to Right).

$$IBM(k,\omega)=\begin{cases}1 & if\ SNR(k,\omega)>LC\\ 0 & if\ SNR(k,\omega)<LC\end{cases} \quad (3)$$

Where $S(k,\omega)$ and $D(k,\omega)$ denote T-F units of the speech and noise energies, whereas LC is local criterion. We found that $LC=0dB$ to be the finest choice for the mask estimation. The estimated magnitude of the clean speech is achieved by multiplying the estimated IBM mask with the noisy speech magnitudes. We have extracted the phase directly from the noisy speech because human auditory system remains unresponsive to small phase errors. Finally, inverse filtering is applied to reconstruct time-domain speech.

Speech enhancement formulates noisy speech signals to enhanced signals with better perceptual quality and intelligibility and usually is considered as estimate of clean speech. Supervised speech enhancement maps this process as a supervised learning problem so that mapping is determined absolutely from the input data. The proposed method contains four modules: feature extraction, training, decoding of DNN and waveform reconstruction. In training stage, DNN model is trained by using features of the noisy and underlying clean speech signals. The acoustic feature sets include the PLP, RASTA-PLP, MFCC, GFCC and AMS. We have selected the combination of RASTA-PLP MFCC and AMS acoustic features. The features are coupled with related delta features. Auto-regressive moving average (ARMA) filter is applied to smooth temporal curves of extracted features to improve speaker identification rates:

$$\hat{F}(t)=\frac{\hat{F}(t-k)+.....+F(t)+.....+F(t+k)}{2k+1} \quad (4)$$

Where $F(t)$ shows the feature vector at time frame t, $\hat{F}(t)$ is filtered feature vector and k is the order of filter. A second order ($k=2$) ARMA filter is used.

A. Network Architecture and Training

The DNN follows the feed-forward structure with five hidden layers, every layer contains 1024 hidden units and 64 output units. Rectified Linear Unit (ReLU) [24] activation functions are used in the hidden layers and also used in output units. ReLU is non-liner in the nature; hence more suitable for speech signals. Additionally, if considering the sparsity of the activation functions, sigmoid or Tanh processes all the neurons, hence make the network dense and costly. On the other hand, ReLU do not activate all the neurons and thereby makes the activations sparse and more efficient. Moreover, ReLU is less computationally expensive than sigmoid and Tanh since it involves simpler mathematical operations. Generally, almost all DNN based speech enhancement use either RBM or autoencoder based pretraining for learning. Yet, for sufficiently large and varied datasets, the pretraining stage can be eliminated and in this paper we use random initialization to initialize DNNs. Additionally, 20% dropout rate is applied to five hidden layers at training stage to decrease the overfitting phenomenon. The adaptive gradient descent (AGD) [25] is coupled with a momentum term κ to optimize the DNNs. For the first initial few epochs, κ rate is fixed at 0.5 but κ rate is increased and fixed at 0.8. The network is trained with mean squared error (MSE), as cost function, for error-correction. The Dropout regularization [26] is used to manage the mismatch conditions. The DNN framework used in this paper is shown in Fig. 3, where H1, H2… are a number of hidden layers. Each hidden layer contains 1024 neurons. Therefore, the total number of neurons in all layers is 5120 which shows a deep neural network.

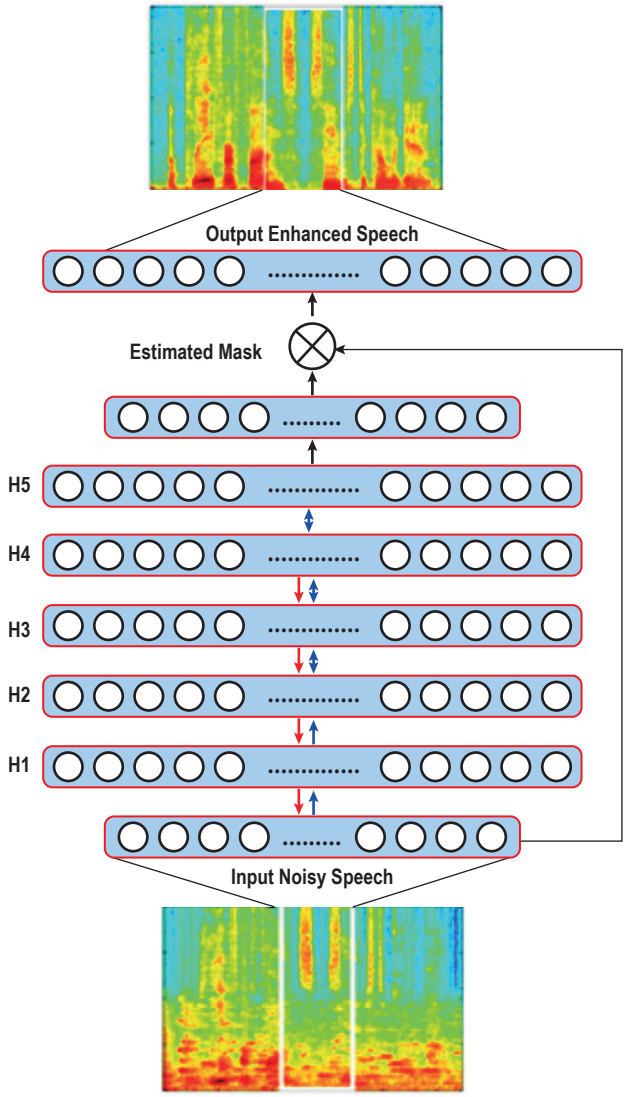


Fig. 3. DNN Training Framework.

III. Experimental Setup

A set of 720 IEEE [27] speech utterances is used during DNN training. The testing set contains 300 speech utterances from unknown speakers of both genders. We have used a real-world environment where multi-noise sources are degrading the target speech utterance. A restaurant like environment is considered in the experiments. The aim of selecting such noisy environment is two-folds: (a) this kind of noisy environment is complex to handle by a speech enhancement algorithm, thus appropriate to test the proposed algorithm; (b) this kind of noise contains diverse noise sources with different power spectral densities shown in Fig. 2. Such noise environment is more practical as individuals are exposed frequently to this noise type. The restaurant noise in our experiments is the mixture of 32-talker multi-babble noise, fan noise and noise originated from striking of the spoons, is supposed to be a real-world environment. The noise source contains both nonstationary (talking people and striking spoons) and stationary (fan noise) conditions. The duration of noise source is approximately 10 minutes.

To build the training set, the first-half of noise source is mixed with training utterances at -5dB, 0dB, 5dB and 10dB SNRs, respectively. The testing mixtures are built by mixing the last half of noise source.

We used five DNN models based on number of the hidden layers, represented by Δ+DNN$_1$, Δ+DNN$_2$, Δ+DNN$_3$, Δ+DNN$_4$ and Δ+DNN$_5$ models. For objective speech quality evaluation, we used Perceptual Evaluation of Speech Quality (PESQ) [28] whereas to evaluate the noise suppression, Segmental SNR (SNRSeg) [29] is used. Short-Time Objective Intelligibility (STOI) [30] is used to predict speech intelligibility. STOI refers to correlation between clean and enhanced signals and has been demonstrated to show high correlation to human speech intelligibility. To examine the DNNs, two distance measures, LLR and WSS, are used. The smaller values of distance measures indicate better result whereas the high values of PESQ, SNRseg and STOI indicate better performance. In our experiments, we have selected Wiener filtering (WF) and non-negative dynamical system (NNDS) as competing methods. The Wiener filtering is an unsupervised approach where NNDS is a supervised method. The DNNs represent a class of supervised methods for speech enhancement. Hence, we have compared the DNN approach with both supervised and unsupervised methods.

IV. Results and Discussions

As declared earlier, the goal of this study is to enhance a noisy speech using DNN in conditions similar to real-world environments. We have selected restaurant-environment for this study. We measured the quality and the intelligibility of the reconstructed enhanced speech in terms of the PESQ and STOI. Table I shows PESQ analysis for noisy, WF, NNDS and DNN with different layers at four input SNRs. The high PESQ scores of DNN show better performance. It is evident that speech quality achieved by DNN with four hidden layers (Δ+DNN$_4$) are higher than the noisy speech, two competing methods and DNN models with three and five hidden layers, that suggests improved speech quality of Δ+DNN$_4$. Table II presents the values of SNRSeg to indicate the suppression capabilities of DNN and other competing methods. Again Δ+DNN$_4$ performed better as compare to Δ+DNN$_3$, Δ+DNN$_5$ and competing methods. The noise is effectively reduced by DNN. Tables III-IV show performance analysis in terms of LLR and WSS. Clearly Tables III-IV indicate that distance between clean and reconstructed speech utterances is less for Δ+DNN$_4$. From Tables I-IV, it is clear that DNN with four hidden layers performed better as compared to DNN with other hidden layers. Therefore, DNN with four hidden layers is suggested to improve the quality and intelligibility of speech degraded by this particular real-time like noise source. The improvements in terms of the PESQ, SNRSeg, LLR and WSS are evident in Tables I-IV.

TABLE I. PESQ Analysis

SNR	Noisy	WF	NNDS	DNN$_3$	DNN$_4$	DNN$_5$
-5	1.57	1.61	1.66	1.68	1.72	1.67
0	1.77	1.94	2.08	2.19	2.22	2.21
5	2.08	2.21	2.39	2.42	2.46	2.42
10	2.46	2.63	2.59	2.65	2.68	2.66
Avg.	1.97	2.10	2.18	2.24	**2.27**	2.24

TABLE II. SNRSeg Analysis

SNR	Noisy	WF	NNDS	DNN$_3$	DNN$_4$	DNN$_5$
-5	0.50	1.24	1.54	1.73	1.78	1.70
0	1.08	2.18	2.98	3.14	3.21	3.20
5	2.30	3.31	4.54	4.93	4.98	4.92
10	4.40	5.43	6.29	6.92	7.02	6.93
Avg.	2.07	3.04	3.83	4.18	**4.24**	4.18

TABLE III. LLR Analysis

SNR	Noisy	WF	NNDS	DNN$_3$	DNN$_4$	DNN$_5$
-5	2.07	1.27	1.01	0.77	0.75	0.76
0	1.61	1.04	0.91	0.86	0.60	0.71
5	1.14	0.78	0.62	0.43	0.33	0.45
10	0.82	0.60	0.59	0.52	0.44	0.40
Avg.	1.41	0.92	0.78	0.64	**0.53**	0.58

TABLE IV. WSS Analysis

SNR	Noisy	WF	NNDS	DNN$_3$	DNN$_4$	DNN$_5$
-5	70.72	57.40	56.01	50.21	48.37	49.07
0	61.02	50.88	48.53	44.52	43.35	44.15
5	53.53	49.31	47.00	42.02	41.22	43.60
10	43.03	37.71	35.54	33.24	31.97	32.60
Avg.	57.07	48.82	46.77	42.49	**41.22**	42.35

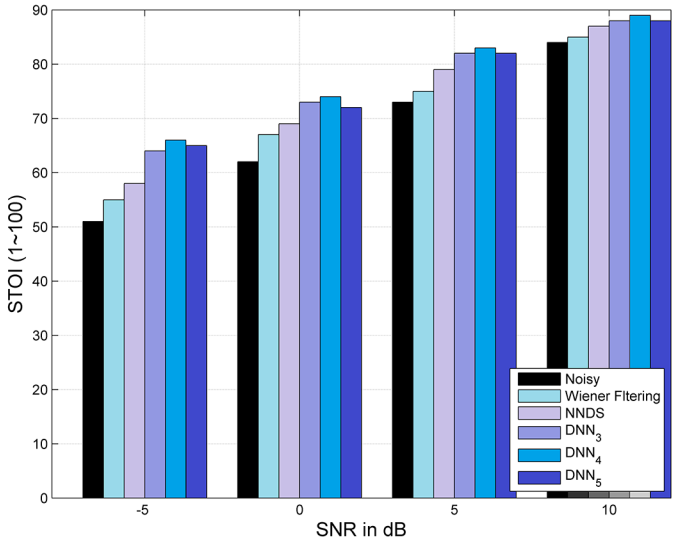


Fig. 4. Intelligibility Analysis.

Speech enhancement in order to improve the speech intelligibility using STOI measures is presented in Fig. 4. The achieved STOI scores with the WF, NNDS and DNN approaches in real-time like condition at four SNRs shows that a better performance is observed with DNN approach than with the two competing approaches and noisy unprocessed speech. Again STOI is computed for three DNN models i.e., Δ+DNN$_3$, Δ+DNN$_4$ and Δ+DNN$_5$ and two competing approaches. It is obvious from Fig. 4 that Δ+DNN$_4$ achieved the highest STOI score as compare to other models and competing approaches. The average STOI score is increased from 50% with noisy speech to 66.2% with Δ+DNN$_4$ at -5dB SNR. Similarly, the average STOI score is increased from 67% with WF to 78% with Δ+DNN$_4$ at 0dB SNR. Moreover, the average STOI score is increased from 79% with NNDS to 84% with Δ+DNN$_4$ at 5dB SNR. By observing the STOI scores, it is evident that DNN with four hidden layers (Δ+DNN$_4$) performed well in improving speech intelligibility. Fig. 5 shows the MSE cost-function values for representing errors at 20 epochs at all input SNRs. We have considered one noise source: 32-talkers babble, for example, and shown that lowest MSE is achieved at 20th epoch. At low SNRs (-5dB and 0dB) a considerable MSE is achieved at epochs greater than 16. For higher SNRs (5dB and 10dB), the MSE is greater as compared to low SNRs.

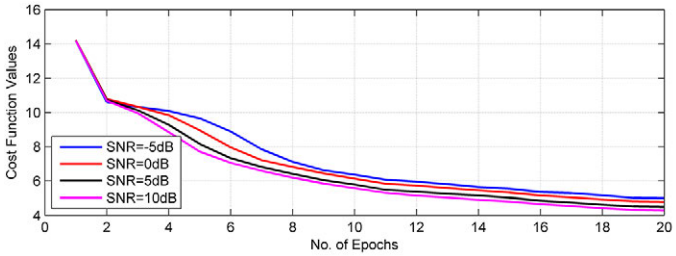


Fig. 5. Objective Cost Function at epochs.

Recently in [31], it has been shown that HIT–FA measure correlates well with human intelligibility. The term HIT indicates the percentage of correctly classified target-dominant T-F units and FA indicates the false alarm or the percentage of wrongly classified interference-dominant T-F units. A fine estimate of IBM ought to have high HIT rate and low FA rate respectively, which guides to high HIT–FA rates. We have used HIT–FA rates in our study to indicate the classification and estimation errors. Fig. 6 shows the HIT-FA rates for the estimated masks for different DNN models. It is evident that Δ+DNN$_4$ has achieved high HIT-FA rates.

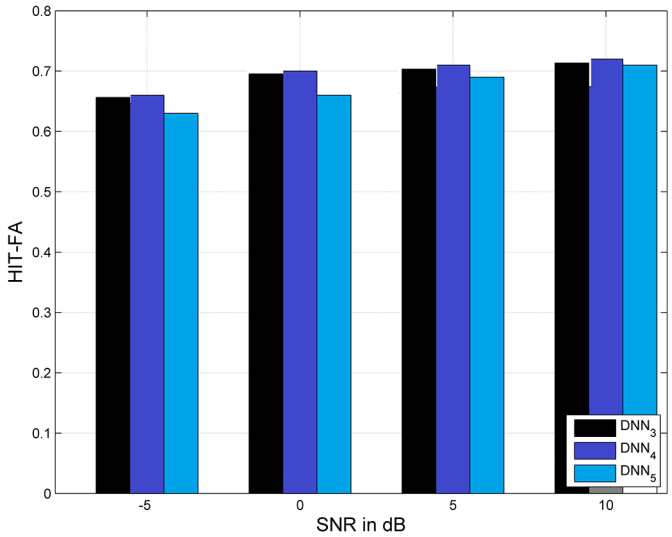


Fig. 6. HIT-FA analysis.

In speech enhancement, distortion is a vital parameter that indicates the ability of understanding a spoken enhanced speech utterance (intelligibility). The distorted utterance would lose vital speech contents, which results in loss of the speech intelligibility. Consequently, it is vital to perform enhancement of noisy speech in such a way that noise is reduced but not at the cost of intelligibility. To examine the speech distortion and residual noise, we have conducted spectrogram analysis. The spectrograms of the enhanced speech obtained with all processing methods are depicted in Fig. 7(A). The spectrograms of WF and NNDS have lost some important speech contents, hence provided less speech intelligibility as compared to DNN model Δ+DNN$_4$ which is evident in the Fig. 7(A). If we note the spectrogram of DNN, we obtained a close replica of clean speech spectrogram and important speech contents are effectively preserved. Also a low residual noise is observed in the spectrogram of DNN output speech. The time-domain waveforms of the enhanced speech utterances obtained with all the processing methods are depicted in Fig. 7(B). The waveforms of WF and NNDS have some residual noise, hence provided less segmental SNR (quality) as compared to DNN model Δ+DNN$_4$ which is evident in the Fig. 7(B). If we observe, the waveform of DNN is a close replica of clean speech waveform and important speech contents are effectively preserved. Also a low residual noise is observed in the waveform of DNN output speech.

V. Summary and Conclusions

This paper considered the restaurant noise problem for speech enhancement which is identical to real-world environments and many noise sources that concurrently degrade quality and intelligibility of a target speech. The existing studies on the speech enhancement principally focus on the presence of one noise source. However, in real-world situations, attempts are made to improve the speech quality and intelligibility of speech where many stationary and nonstationary noise sources are simultaneously mixed with target speech. To address such problem, we have used Deep Neural Networks approach and used ideal binary mask (IBM) as a binary classification method and target function during training. The mean square error (MSE) objective cost function is used during training to reduce errors. The experimental results at different input SNRs have confirmed the superiority of DNN-based multi-noise speech enhancement in terms of PESQ, SNRSeg, LLR, WSS and STOI. Our experimental results in particular noisy situations have demonstrated an average 7% improvement in speech quality as compared to noisy speech. Similarly, an average 6.5% improvement in speech intelligibility is noted during experiments. Moreover, a large improvement in terms of the SNRSeg, LLR and WSS is recorded during experiments, shown in Tables II-IV for reference. At low SNRs (-5dB) the DNN based speech enhancement in this particular noise source performed exceptionally and attained large improvements. The time-varying spectral analysis confirmed that the DNN with four hidden layers has the capacity to reduce considerable noise and the speech contents are preserved to an acceptable level of understanding. The overall analysis of DNN architecture has validated that Δ+DNN$_4$ has a great potential to deal this noise type as compared to other two competing unsupervised and supervised methods.

VI. Future Work

The greater part of the speech processing algorithms operate only with the spectral magnitude, leaving spectral phase unstructured and unexplored. With recent advancement in deep neural networks, the phase processing became more important as an innovative and emergent prospective of the DNN based speech enhancement. The authors will develop the DNN with phase estimation in future to test the speech intelligibility and quality potentials in the complex noisy environments.

References

[1] Rehr, R., & Gerkmann, T. (2018). On the importance of super-Gaussian speech priors for machine-learning based speech enhancement. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 26(2), 357-366.

[2] Saleem, N., Khattak, M. I., & Shafi, M. (2018). Unsupervised speech enhancement in low SNR environments via sparseness and temporal gradient regularization. Applied Acoustics, 141, 333-347.

[3] Saleem, N., Irfan, M., Chen, X., & Ali, M. (2018). Deep Neural Network based Supervised Speech Enhancement in Speech-Babble Noise. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS) (pp. 871-874). IEEE.

[4] Saleem, N., Shafi, M., Mustafa, E., & Nawaz, A. (2015). A novel binary mask estimation based on spectral subtraction gain-induced distortions for improved speech intelligibility and quality. University of Engineering and Technology Taxila. Technical Journal, 20(4), 36.

[5] Saleem, N., & Irfan, M. (2018). Noise reduction based on soft masks by incorporating SNR uncertainty in frequency domain. Circuits, Systems,

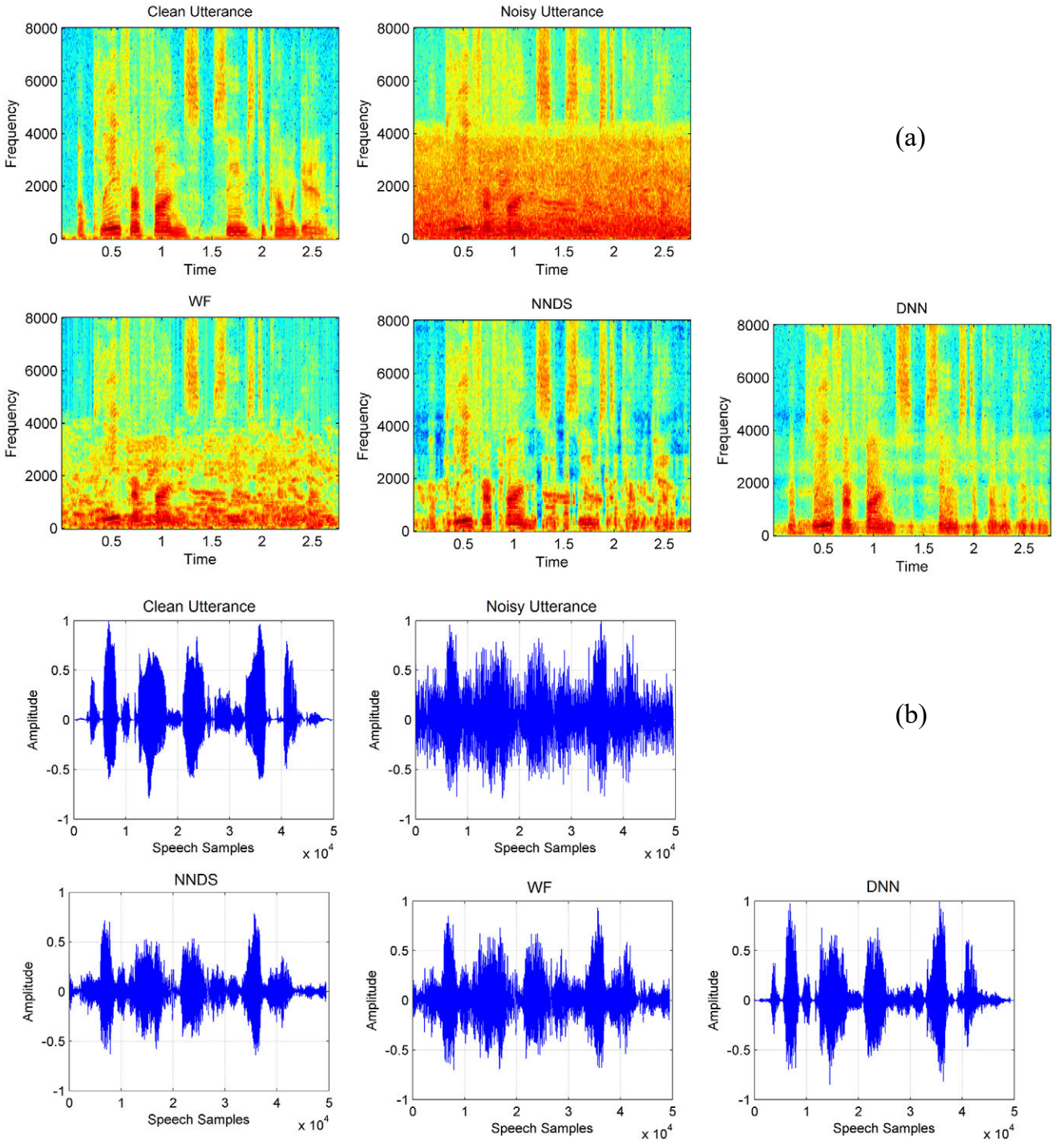


Fig. 7. Spectrogram and Waveform Analysis.

and Signal Processing, 37(6), 2591-2612.

[6] Saleem, N., & Khattak, M. I. (2018). Regularized sparse decomposition model for speech enhancement via convex distortion measure. Modern Physics Letters B, 32(22), 1850262.

[7] Saleem, N., & Tareen, T. G. (2018). Spectral Restoration Based Speech Enhancement for Robust Speaker Identification. International Journal of Interactive Multimedia and Artificial Intelligence, 5(1), 34-39.

[8] Boll, S. (1979, April). A spectral subtraction algorithm for suppression of acoustic noise in speech. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'79. (Vol. 4, pp. 200-203). IEEE.

[9] Scalart, P. (1996, May). Speech enhancement based on a priori signal to noise estimation. In Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings, 1996 IEEE International Conference on (Vol. 2, pp. 629-632). IEEE.

[10] Ephraim, Y., & Malah, D. (1984). Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. IEEE Transactions on acoustics, speech, and signal processing, 32(6), 1109-1121.

[11] Ephraim, Y., & Malah, D. (1985). Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE transactions on acoustics, speech, and signal processing, 33(2), 443-445.

[12] Ephraim, Y., & Van Trees, H. L. (1995). A signal subspace approach for speech enhancement. IEEE Transactions on speech and audio processing, 3(4), 251-266.

[13] Sigg, C. D., Dikk, T., & Buhmann, J. M. (2010, March). Speech enhancement with sparse coding in learned dictionaries. In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on (pp. 4758-4761). IEEE.

[14] Zao, L., Coelho, R., & Flandrin, P. (2014). Speech enhancement with emd and hurst-based mode selection. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(5), 899-911.

[15] Deng, L., Hinton, G., & Kingsbury, B. (2013, May). New types of deep neural network learning for speech recognition and related applications: An overview. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on (pp. 8599-8603). IEEE.

[16] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine, 29(6), 82-97.

[17] Xu, Y., Du, J., Dai, L. R., & Lee, C. H. (2014). An experimental study on speech enhancement based on deep neural networks. IEEE Signal processing letters, 21(1), 65-68.

[18] Xu, Y., Du, J., Dai, L. R., & Lee, C. H. (2015). A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 23(1), 7-19.

[19] Kolbk, M., Tan, Z. H., Jensen, J., Kolbk, M., Tan, Z. H., & Jensen, J. (2017). Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 25(1), 153-167.

[20] Lu, X., Tsao, Y., Matsuda, S., & Hori, C. (2013, August). Speech enhancement based on deep denoising autoencoder. In Interspeech (pp. 436-440).

[21] Wang, Y., Narayanan, A., & Wang, D. (2014). On training targets for supervised speech separation. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 22(12), 1849-1858.

[22] Chen, J., Wang, Y., & Wang, D. (2014). A feature study for classification-based speech separation at low signal-to-noise ratios. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(12), 1993-2002.

[23] Brown, A. M., Gaskill, S. A., Carlyon, R. P., & Williams, D. M. (1993). Acoustic distortion as a measure of frequency selectivity: relation to psychophysical equivalent rectangular bandwidth. The Journal of the Acoustical Society of America, 93(6), 3291-3297.

[24] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 807-814).

[25] Klein, S., Pluim, J. P., Staring, M., & Viergever, M. A. (2009). Adaptive stochastic gradient descent optimisation for image registration. International journal of computer vision, 81(3), 227.

[26] Wager, S., Wang, S., & Liang, P. S. (2013). Dropout training as adaptive regularization. In Advances in neural information processing systems (pp. 351-359).

[27] Rothauser, E. H. (1969). IEEE recommended practice for speech quality measurements. IEEE Trans. on Audio and Electroacoustics, 17, 225-246.

[28] Rix, A. W., Beerends, J. G., Hollier, M. P., & Hekstra, A. P. (2001). Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP'01). 2001 IEEE International Conference on (Vol. 2, pp. 749-752). IEEE.

[29] Loizou, P. C. (2007). Speech enhancement: theory and practice. CRC press.

[30] Taal, C. H., Hendriks, R. C., Heusdens, R., & Jensen, J. (2010, March). A short-time objective intelligibility measure for time-frequency weighted noisy speech. In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on (pp. 4214-4217). IEEE.

[31] Kim, G., Lu, Y., Hu, Y., & Loizou, P. C. (2009). An algorithm that improves speech intelligibility in noise for normal-hearing listeners. The Journal of the Acoustical Society of America, 126(3), 1486-1494.

Nasir Saleem

Engr. Nasir Saleem received the B.S degree in Telecommunication Engineering from University of Engineering and Technology, Peshawar-25000, Pakistan in 2008 and M.S degree in Electrical Engineering from CECOS University, Peshawar, Pakistan in 2012. He was a senior Lecturer at the Institute of Engineering and Technology, Gomal University, D.I.Khan-29050, Pakistan. He is now an Assistant Professor in the Department of Electrical Engineering, Gomal University, Pakistan. His research interests are in the area of digital signal processing, speech processing and speech enhancement.

Muhammad Irfan Khattak

Muhammad Irfan Khattak is working as an Associate Professor in the Department of Electrical Engineering in the University of Engineering and Technology Peshawar. He did his B.Sc Electrical Engineering from the same University in 2004 and did his PhD from Loughborough University UK in 2010. His research interest involves Antenna Design, On-Body Communications, Speech processing and Speech Enhancement.

Binary Multi-Verse Optimization (BMVO) Approaches for Feature Selection

Rahul Hans[1,2], Harjot Kaur[1] *

[1] Department of Computer Science and Engineering, Guru Nanak Dev University, Regional Campus, Gurdaspur, Punjab (India)
[2] Department of Computer Science and Engineering, DAV University, Jalandhar, Punjab (India)

Received 19 October 2018 | Accepted 11 May 2019 | Published 24 July 2019

Abstract

Multi-Verse Optimization (MVO) is one of the newest meta-heuristic optimization algorithms which imitates the theory of Multi-Verse in Physics and resembles the interaction among the various universes. In problem domains like feature selection, the solutions are often constrained to the binary values viz. 0 and 1. With regard to this, in this paper, binary versions of MVO algorithm have been proposed with two prime aims: firstly, to remove redundant and irrelevant features from the dataset and secondly, to achieve better classification accuracy. The proposed binary versions use the concept of transformation functions for the mapping of a continuous version of the MVO algorithm to its binary versions. For carrying out the experiments, 21 diverse datasets have been used to compare the Binary MVO (BMVO) with some binary versions of existing meta-heuristic algorithms. It has been observed that the proposed BMVO approaches have outperformed in terms of a number of features selected and the accuracy of the classification process.

Keywords

Machine Learning,
Feature Selection,
K-Nearest Neighbors,
Binary Multi-Verse Optimization.

DOI: 10.9781/ijimai.2019.07.004

I. Introduction and Rationale

In the current era, due to swift advancement in the field of Computer Science, an enormous amount of information is generated and is included in the datasets in the form of a large number of features. In Machine learning, a feature is defined as the individual assessable quality of the process being observed [1]. However, the datasets may consist of a huge number of redundant features and the features which may not be relevant to the target concept [2]. These irrelevant features require more storage and lead to increase the computational cost. Thus, in order to reduce the computational constraint and storage requirement, as well as to improve the classification accuracy, feature selection has become a vital task for complex or large datasets [1] [3].

In literature, there is an enormous amount of machine learning applications where feature selection has been applied. Some of these applications involve medical diagnosis [4], facial expression recognition [5], diagnose of bronchitis [6], gene selection and cancer classification [7], image steganalysis [8], big data classification [9], obstructive sleep apnea diagnosis [10], sentiment classification [11], Mobile Agent Platform Protection [70], Irony Detection[71], categorize text documents [72], classification of Plant Diseases [73], Breast Masses Detection [74].

A. Motivation

The fundamental objectives of the feature selection are to reduce the dimensionality of the data and to improve the prediction performance [17], or in other words to employ fewer features to represent data without fading its discriminative capability. Pondering over, feature selection also aids in avoiding over fitting, resisting noise and improving classification accuracy [12]. This is the reason why finding the optimal subset of features to solve a task has become a necessary practice in machine learning. However, the complexity of task makes it impractical to find the promising solutions in a good enough time. Furthermore, searching for an optimal feature subset from a high dimensional feature space can be considered as an NP-complete problem [15].

In recent times, a class of algorithms called nature-inspired algorithms has drawn to the attention of the researchers to find the solution of optimization problems, nature-inspired algorithms have been developed by drawing inspiration from nature. To get to the bottom of the problem of feature selection, the scope of this research is to investigate the use of one of the most recent nature-inspired optimization algorithms viz. the Multi-Verse Optimization algorithm, to discover the best feature subset combination in diverse datasets (belonging to different domains) leading to maximum classification accuracy and a minimum number of selected features in the feature set. In view of the stochastic nature of the metaheuristic algorithms, metaheuristic algorithms discover the optimal solution by evaluating the sub solutions by invoking a fitness function. The sub solutions are formed by randomly selecting some of the features and rejecting the other features. For that reason, the conversion of the continuous version of the algorithm to its corresponding binary version becomes a significant task. Furthermore, this research aims to explore the use of two different categories of transfer functions viz. S-Shaped and V-Shaped transfer functions, for position updating step of the optimization algorithm for the conversion of the continuous version of the algorithm to its corresponding binary versions.

* Corresponding author.
E-mail addresses: rahulhans@gmail.com (R. Hans), harjotkaursohal@rediffmail.com (H. Kaur).

B. Feature Selection and Optimization

The process of feature selection can be categorized into two different approaches: (i) wrapper-based and (ii) filter-based, that weighs up the eminence of the selected features. The former approach uses a machine learning method to search through the space of potential solutions, whereas, the latter approach goes over the feature space based on the data-dependent norm, unlike former which is based on the classification-dependent norm [13]. Furthermore, out of two, the former algorithms obtain better results; because they consider the association among the learning algorithm and the training data. However, they are computationally costly [3] and slower than the latter algorithms because the learning algorithm executes over and over again for every chosen feature subset [14].

Clearly, the Search process of finding an optimal feature subset from the original set is an exigent task as the complication of the problem of concern makes it impracticable to search every possible solution, so the prime aim is to find the best feasible solution in an acceptable time, which can be defined as an optimization problem [15].

C. Meta-heuristic Algorithm and MVO

In recent years, a class of algorithms known as metaheuristics has been considered more reliable when solving various optimization problems [16]. Metaheuristic algorithms use trial and error approaches to produce acceptable solutions to complex problems in a reasonable time [15]. Basically, these algorithms require balance in two following important components: Exploration of the search space (diversification) and exploitation of the best solutions found during exploration (intensification). In other words, the diversification produces the dissimilar solutions so as to explore the search space on a global scale, while intensification concentrates on the search in the local region by making use of information that a current good solution is found in this region. Meta-heuristic algorithms can also be divided into two classes: population-based algorithms (for instance, swarm intelligence, evolutionary algorithm) which are exploration-oriented and single-solution based algorithms (for instance, local search, simulated annealing) which are exploitation-oriented [16]. Last decade has revealed numerous efforts from researchers (as shown in Table I), who has reaped the benefits of metaheuristic algorithms to get to the bottom of the feature selection problem. Multi-Verse Optimization (MVO) is a newly introduced metaheuristic algorithm, which is inspired by the Multi-Verse theory in Physics. In this paper, the binary versions of MVO have been proposed and applied to solve one of the important problems of machine learning namely feature selection. The proposed binary versions of MVO use the concept of transformation functions for mapping of continuous version of MVO to the binary version of MVO algorithm.

D. Organization of the Paper

The rest of this paper is structured as follows: the brief analysis of previous related research works is presented in Section II. Section III throws light upon the concept of Multi-Verse Optimization (MVO) algorithm, whereas Section IV puts forward the binary versions of the MVO algorithm (BMVO) by using the concept of transformation functions. Furthermore, in Section V, various materials and methods used for the proposed study are discussed. Section VI briefly presents the results and discussions. Finally, in Section VII, conclusions and future works are presented.

II. The Current State-of-the-Art in Metaheuristic Algorithms

In the past many years, flourishing work has been accomplished on selecting an optimal number of features that are likely to preserve higher classification accuracy rate, by means of metaheuristic algorithms. In this regard, this section aims to analyze the use of various metaheuristic algorithms viz. Ant colony optimization(ACO), Particle Swarm Optimization(PSO), Firefly Optimization Algorithm(FOA), Artificial Bee Colony algorithm(ABC), Genetic Algorithm(GA), Grey Wolf Optimization(GWO), Binary Bat Algorithm (BA), Simulated Annealing (SA), Ant lion Optimization (ALO), Cuckoo Search (CS), Sine Cosine Optimization Algorithm(SCA), Differential Evolution(DE) ,Whale Optimization algorithm(WOA) and their variants. The summary of various metaheuristic algorithms and their variants explored in recent years for feature selection has been given in Table I.

In [14] Manizheh Ghaemi, et al. have proposed the use of Forest Optimization Algorithm (FOA) for feature subset selection which was tested on 11 different datasets and compared with GA, PSO, and ACO. Uros Mlakar, et al. in [5], have used the notion of Multi-Objective DE for feature selection in Facial Expression Recognition. Similarly, Indu Jain et. al in [7], have proposed an improved-Binary Particle Swarm Optimization with Correlation-based Feature Selection (CFS) which uses the scheme of increasing inertia weight and controls the searching capability of the iBPSO algorithm. The performance of iBPSO has been evaluated using 11 benchmark microarray datasets of different cancer types and the results have been compared with methods like SVM, Random Forest, Fast Correlation-Based Filter, BPSO, PSO-DT.

In [17] Chuang et al. proposed an improved binary PSO for feature selection for gene expression data classification problems. By returning the gbest, improved binary PSO can avoid getting trapped in a local optimum, and better classification result can be attained with a lesser amount of chosen genes. Results also demonstrate that improved binary PSO efficiently simplifies feature selection and reduces the number of features desired. The algorithm was tested on 11 gene expression data test problems and the proposed method has the highest classification accuracy in 9 of the 11 gene expression data test problems.

In [18] Yu-Peng Chen et al. have proposed the two novel variants of Bacterial foraging optimization (BFO) algorithms viz. Adaptive Chemotaxis Bacterial Foraging Optimization Algorithm (ACBFO) and Improved Swarming and Elimination-Dispersal Bacterial Foraging Optimization Algorithm (ISEDBFO).The proposed algorithms have been tested on 10 different benchmarks datasets and performance has been compared with PSO, GA, SA, ALO, BBA, CS. Also, J. Dhalia Sweetlin et al. in [6] have proposed the combination of ACO with cosine similarity and Support Vector Machine (SVM). The outcome of the proposed algorithms is compared with PSO and hybrid PSO algorithms. GA is used by Babatunde Oluleye et al. in [19] in combination with the classifiers of weka (MLP, RF, J48, NB, RC) and weka feature selection algorithms (Correlation Feature Selection Subset Evaluator, Information gain). For evaluating the performance two different datasets are used viz. features extracted from Flavia dataset and ionosphere dataset.

Pedram Ghamisi et.al in [20] have proposed the hybridization of GA and PSO. The accuracy of SVM classifier obtained on validation samples is used as a fitness value and it is evaluated on the Indian Pines hyperspectral data set. Gang Wang et al., in [21] have explored the use of ACO by adaptively adjusting its parameters (such as pheromone evaporation rate, number of ants and exploration probability factor) for feature selection. The results are evaluated on 10 different datasets and performance is compared with GA, PSO, ACO, fuzzy adaptive ant system. In [22] Fadzil Ahmad, et al. have used GA for simultaneous feature selection and parameter optimization of an artificial neural network (ANN) and for feature selection the performance of resilient back-propagation (GAANN_RP) with and without feature selection is compared on breast cancer dataset. H. Hannah Inbarani et al. in [4] applied PSO integrating with a rough set theory for feature subset selection and compared the performance on 4 different datasets.

TABLE I. Brief Summary of Metaheuristic Algorithms and their Variants Explored in Recent Years for Feature Selection

Authors	Year	Algorithm	Strategy Used	No. of Datasets	Algorithms for Comparison
Li-Yeh Chuang, et al. [30]	2009	Particle Swarm Optimization	Use of Chaotic maps with PSO	10	BPSO
Hema Banati,Monika Bajaj [31]	2011	Fire fly Algorithm	Fire fly with rough set theory	4	RSAR*,AntRSAR,GenRSAR, PSORSAR,BeeRSAR,FA_RSAR
Li-Fei Chen et al. [10]	2011	Particle swarm optimization	Integrating Particle swarm optimization with 1-Nearest Neighbour scheme	8	BPN LR SVM C4.5 GA + 1-NN PSO + 1-NN
Yuanning Liu, et al. [32]	2011	Particle Swarm Optimization	Employed sub-swarms and a multi-swarm scheduler which monitors each sub-swarm	10	PSO+SVM and GA+SVM
R. Y. M. Nakamura, et al. [33]	2012	Binary Bat Algorithm	Used Sigmoid function to restrict the new bat's position to only binary values	5	PSO, FFA, HS, GSA
XiaoHong Han, et al. [34]	2014	Gravitational search algorithm	Use of Piecewise linear chaotic map for increasing diversity, SQP* for increasing local exploitation	15	BPSO, GA
E. Emary, et al. [35]	2015	Firefly Algorithm	Attraction is randomized with α being the randomization parameter	18	GA, PSO
Hossam M. Zawbaa, et al. [13]	2016	Moth-flame optimization	-	18	GA, PSO
E. Emary, et al. [36]	2016	Gray Wolf Optimization	Use of Sigmoid function and Crossover technique	18	GA, PSO
Lin Shang, et al. [11]	2016	Particle swarm optimization	Used Fitness proportionate selection binary particle swarm optimization	2	BPSO F-BPSO FS-BPSO
Hong Wang, Ben Niu [37]	2016	Bacterial algorithm	It uses modified population updating strategies and control mechanisms.	10	BFO, BFO-LDC,FO-NDC, BCO, BAFS
Majdi M. Mafarja and Seyedali Mirjalili [16]	2017	Whale Optimization Algorithm	Whale Optimization Algorithm with simulated annealing	18	ALO, GA, PSO
Gehad Ismail Sayed, et al. [3]	2017	Crow Search	Use of Chaotic Maps	20	PSO, ABC, CSO, FPA, MFO, GWO, etc
Majdi Mafarja, et al. [38]	2017	Ant loin Optimization	Use of s-shaped and v-shaped functions	18	GSA, PSO
Ahmed A. Ewees, et al. [39]	2017	Multiverse Optimization	Use of Chaotic Maps	5	MVO, PSO, ABC
P. Shunmugapriya, S. Kanmani [40]	2017	Ant Colony optimization and Bee colony optimization	Ebees perform exploitation on the solutions generated by ACO	13	ABC-DE Hybrid, AC-ABC Hybrid
Mohammed Aladeemy, et al. [41]	2017	Self-Adaptive Cohort Intelligence	For the sampling interval and mutation rate, It employed tournament-based mutation and self-adaptive scheme	10	SVM-CI, SVM-ABC, SVM, GA, SVM-DE, and SVM-PSO
E. Tamimi [61]	2017	ACO GA ICA PSO	Optimization of SVM	2	Random forest
Sushama Nagpal, et al. [42]	2017	Gravitational search algorithm	GSA and k-nearest neighbors	3	GA, PSO, GSA
M. K Sohrabi et al. [59]	2017	Non-dominated Sorting GA-II and Multi-Objective PSO	Non-dominated Sorting GA-II and Multi-Objective PSO and ANN	1	Classical methods
Hossam Faris, et al. [43]	2017	Multi verse Optimization	MVO for selecting optimal features and optimizing the parameters of SVM	10	GA PSO BAT FF
Ibrahim Aljarah, et al. [44]	2017	Grasshopper optimization algorithm	Optimizing SVM based on the Grasshopper Optimization Algorithm (a hybrid approach)	12	GOA Grid search MVO GA PSO GSA
Majdi Mafarjaa, Seyedali Mirjalili [45]	2017	Whale optimization	Tournament and Roulette Wheel selection mechanisms instead of random operator	20	ALO GA PSO
M. Suganthi ,V. Karunakaran [46]	2017	Cuttlefish optimization algorithm	In this PCA* is used for feature extraction with Cuttlefish optimization	4	Algorithm with Instance selection, Algorithm with Feature extraction
D. Zouache et al. [60]	2018	FF and PSO	Quantum computation/rough set theory	11	PSO-Rough Set, FSA-Rough Set
Hongbin Dong, et al. [47]	2018	Genetic Algorithm	Neighborhood policy to granulate the sample space and improved neighborhood rough set	11	FCBF, BIRS, MBEGA, GA, mRMR, IBGAFG, INRSG
E.Emary , et al. [67]	2018	Ant lion optimization	Levy flights with Ant lion Optimization	21	GA PSO ALO LALO

RSAR* - Rough Set Based Attribute Reduction, SQP*- Sequential quadratic programming, PCA*- Principal component analysis

Similarly, Yumin Chen et al. in [23] have explored the use of ACO with rough sets and evaluated the performance on 9 different datasets.

S. Tiwari et al. in [62] used local searching algorithms for spawning relevant and non-redundant features; subsequently, a global optimization algorithm has been used to remove the restrictions of global optimization algorithms. The time and accuracy were improved by using a feature set obtained from sequential backward selection and mutual information maximization algorithm which is fed to a global optimization technique (GA, PSO etc). To test the proposed approach publicly available Sonar, Wdbc and German datasets were used.

A. Ekbal in [63] proposed an algorithm based on the notion of multiobjective optimization viz. multiobjective GA, along with the lines of NSGA-II, for performing parameter optimization and feature selection. The experiments were performed on four different classifiers viz. random field, support vector machine, memory based learner and maximum entropy. The proposed algorithms are evaluated for solving the problems of named entity recognition (used in text processing applications). Comparisons with existing researches demonstrated the usefulness of the proposed approach.

Shih-Wei Lin et al. in [64] proposed a particle swarm optimization based approach to obtain the suitable parameter settings for the back-propagation network and to select the relevant subset of features which resulted in improved accuracy. For experimental evaluation of the proposed algorithm 23 different datasets were considered from the UCI repository. When the performance was compared with some existing works, the results confirmed that the proposed approach improved the classification accuracy in the majority of test problems.

J. Vijaya in [65] proposed an algorithm for the telecom churn prediction that makes use of particle swarm optimization and proposes three different versions of PSO for churn prediction that is, PSO having feature selection as its pre-processing mechanism, PSO with simulated annealing and lastly PSO with a blend of both feature selection and simulated annealing.

R. P. S. Manikandan et al. in [66] proposed fish swarm optimization for feature selection in big data. To perform the experiments, Product review dataset that is obtained from Amazon along with synthetic data is used. The dataset contained 235,000 positive and 147,000 negative reviews. Results depicted that the technique proposed attains improved performance than that of the other techniques.

III. Multiverse Optimization Algorithm (MVO)

Multi-verse is another latest hypothesis among researchers in physics [68]. MVO algorithm is based on the ideas of white holes, black holes, and wormholes which are mathematically mapped to construct the MVO [15]. The word Multi-Verse implies "opposite of universe", which means there exists one more universe besides the one in which we are living [13].

A. Notion

It is believed that more than one big bang happenings have taken place and each big bang has led to the dawn of a new universe. This theory also states that numerous universes act together and might even have a collision with each other. Furthermore, it puts forward a fact that there might be different physical laws in each of the universes.

Significantly, there are following three concepts of the multiverse theory which has inspired the MVO the most:

- White holes- these can be accounted for as a big bang which may be the source of the origin of the universe [24].
- Black holes- these are assumed as to pull towards them every object with their gravitational force [25].
- Wormholes- these can be assumed as time/space travel channels, where objects are able to travel instantaneously across the universe [26].

It has been assumed that every universe expands through the space caused by its inflation rate [27][28]. Moreover, it has been also argued that multiple universes interact via these three different holes in order to target a stable situation [27][29].

B. Continuous Version of Multi-Verse Optimization Algorithm(MVO)

As discussed in sub-section III.A, in MVO algorithm black holes receive the objects that are transmitted by white holes. So as the number of iterations add to, the fitness (inflation rates) of all the universes will become better. Improvement in the exploration and exploitation phase and to prevent getting rapt in local optima, preservation of the diversity of universes is obligatory which is made possible by wormholes [39]. To begin with MVO algorithm, random universes are generated, and in each iteration, objects are transferred through white and black holes from the universe with higher fitness to the universe with lower fitness by a random transmission via wormholes in the direction of the best universe. This process occurs until the end criterion is not satisfied.

In the MVO algorithm, the following set of laws is implemented on the universes:

1. More the inflation rate more is the likelihood of having a white hole and less is the likelihood of encompassing black holes.
2. High inflation rated universes have a tendency to propel objects through white holes while low inflation rated universes have a tendency to accept objects through black holes.
3. Despite the inflation rate, the objects in the universes face random movement towards the best universe.

To map the algorithm mathematically the white and black holes are represented as the population of universes U. It is supposed that each candidate solution is analogous to a universe and each variable in the solution is assumed as an object β_i^j (j^{th} parameter of the i^{th} universe) in that universe, the dimension of the problem is represented by d and the number of universes is represented by N. The different universes are arranged on the basis of their fitness values (inflation rates). Probing further, the object β_i^j is swapped by using the concept of Roulette Wheel mechanism which selects universe U_k as in equation (1), where r1=[0,1] is a random number, U_i is the i^{th} universe, and NF(U_i) is the normalized fitness value(inflation rate) of the U_i. This complete process of MVO is depicted in Fig. 1.

The smaller value of fitness specifies the more likelihood of transferring objects through white and black holes. By assuming that each U_i transports the objects of U_i in the course of space arbitrarily using wormholes, the exploitation is carried out and the assortment of universes is retained. The objects of the universes are altered by the wormholes in a stochastic manner, where wormhole existence probability (WEP) is defined in equation (3). These wormholes are also used to bring up to date the universe's objects and perk up the inflation rate by altering the objects of the universe which has the finest inflation rate as the subsequent shown in equation (2) [39], where β_j shows the j^{th} parameter of best universe formed so far, lb_j and ub_j indicate the lower bound and the upper bound respectively corresponding to the j^{th} variable r2,r3,r4 are the random numbers between 0 and 1. The coefficient, Travelling Distance Rate (TDR) is employed to choose the distance letting a wormhole move the object towards the best universe, as shown in equation (4),

$$\beta_i^j = \begin{cases} \beta_k^j, & if\ r1 < NF(U_i) \\ \beta_i^j, & if\ r1 > NF(U_i) \end{cases} \quad (1)$$

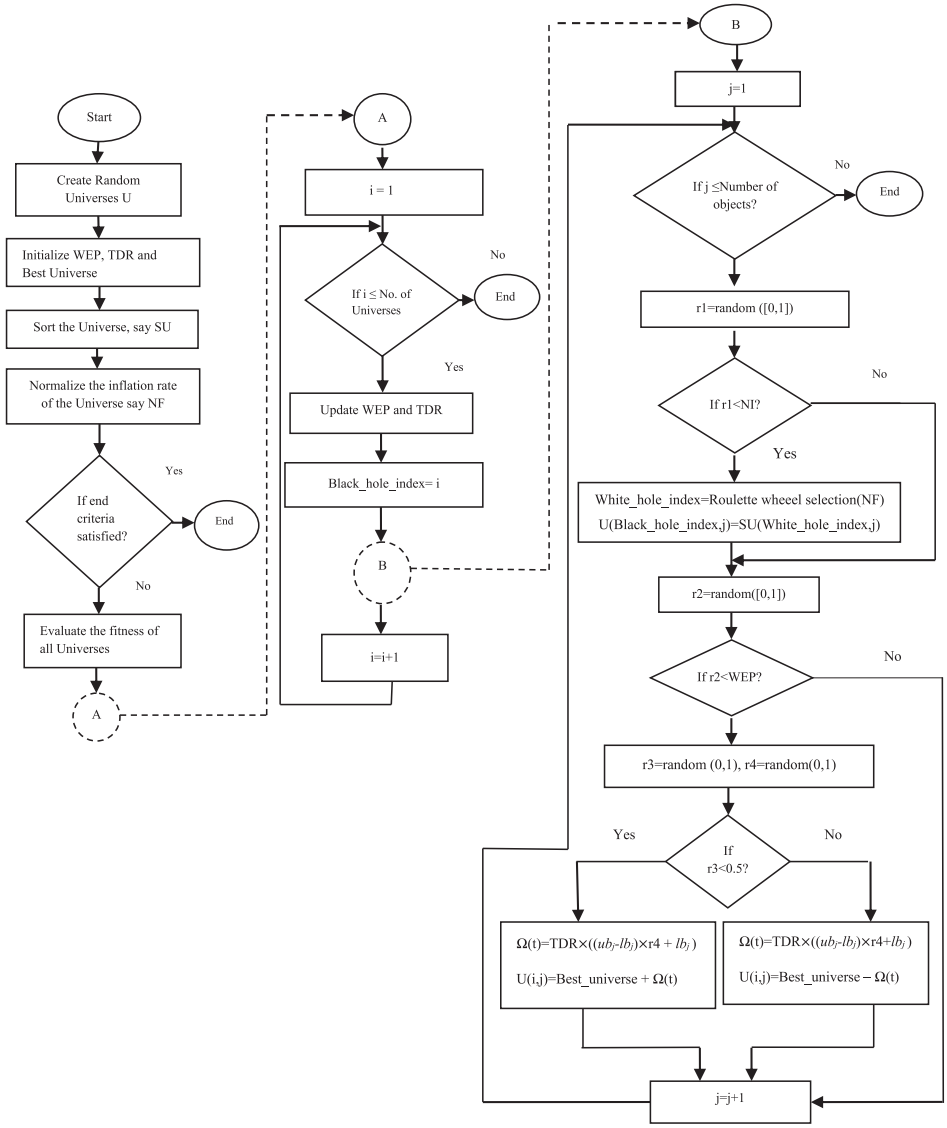


Fig. 1. General Flowchart representation of Multi-verse Optimization algorithm.

$$\beta_i^j = \begin{cases} \beta_j + \text{TDR} \times ((\text{ub}_j - \text{lb}_j) \times r4 + \text{lb}(j)) \geq 1 & r3 < 0.5 \\ \beta_j - \text{TDR} \times ((\text{ub}_j - \text{lb}_j) \times r4 + \text{lb}(j)) \geq 1 & r3 \geq 0.5 \\ \beta_i^j & r2 > \text{WEP} \end{cases} \quad r2 < \text{WEP} \tag{2}$$

$$WEP = MIN_{WEP} + l \times \frac{Max_{WEP} - MIN_{WEP}}{L} \tag{3}$$

$$TDR = 1 - \frac{l^{1/p}}{L^{1/p}} \tag{4}$$

Where l and L defines the current iteration and a total number of iterations respectively. The default value of p is 6. In equation 3, the default minimum(MIN_{WEP}) and maximum(MAX_{WEP}) values of WEP are 0.2 and 1 respectively.

IV. Binary Versions of the Multi-verse Optimization Algorithm (BMVO)

One of the obligatory requirements to find the feature subset containing a minimum number of features and achieving high classification accuracy from those selected features is to transform the continuous version of MVO to binary version. In this transformation, the location of search agents, as well as the position of the search agents, is represented in the form of strings of 0s and 1s. A transformation function defines the likelihood of altering a position vector's elements from 0 to 1 and vice versa [48]. This section examines the functionality and performance of two different V-shaped and two different S-shaped transformation functions for developing different binary versions of MVO [49] [57].

A. Proposed Approach

In the problems with search space consisting of continuous values can be transformed into binary problems by transforming their variables to binary [69]. Irrespective of the binary problem category, the search space of binary problem consists of binary values which may have its own structure. In general, the search space of a binary problem can be represented as a hypercube. So, by flipping some of the bits of the candidate solution (universe/agent) the solution of a binary optimization algorithm may reallocate to closer and farther points of the hypercube [48]. Therefore, to design a binary MVO, this research customizes the position updating procedure.

In binary search space, while dealing with 0s and 1s, equation (2) of MVO cannot perform the position updating directly. Therefore, another way has been proposed to use the equation. (2) for changing agent's location from 1 to 0 or from 0 to 1. In BMVO, the agent's position is transformed with the probability of position updating value (β_i^j) of the objects of the universe, depending upon random number r3. In this regard, four different transformation functions have been explored in this work to map equation (2) values to probability values for updating the agent's position.

1. Binary Mapping Using S-Shaped Family of Transfer Functions

a) *BMVO-I uses Sigmoidal Transformation*

$$\beta_i^j = \begin{cases} 0, & trf1(\beta_i^j) > \theta_1 \\ 1, & otherwise \end{cases} \tag{5}$$

$$trf1\ (\beta_i^j) = sigmoid(\beta_i^j) = \frac{1}{1+ e^{-\beta_i^j}} \tag{6}$$

b) *BMVO-II uses another version of sigmoidal Transformation*

$$\beta_i^j = \begin{cases} 0, & trf2(\beta_i^j) > \theta_2 \\ 1, & otherwise \end{cases} \tag{7}$$

$$trf2(\beta_i^j) = M\_sigmoid(\beta_i^j) = \frac{1}{1+ e^{-(\beta/2)_i^j}} \tag{8}$$

2. Binary Mapping Using V-Shaped Family of Transfer Functions

a) *BMVO-III uses the tangent hyperbolic transfer function*

$$\beta_i^j = \begin{cases} 0, & trf3\ (\beta_i^j) > \theta_3 \\ 1, & otherwise \end{cases} \tag{9}$$

$$trf3\ (\beta_i^j) = tanh(\beta_i^j) = \left| \frac{(e^{-(\beta)_i^j} - 1)}{(e^{-(\beta)_i^j} + 1)} \right| \tag{10}$$

b) *BMVO-III uses the inverse tangent transfer function*

$$\beta_i^j = \begin{cases} 0, & atan(\beta_i^j) > \theta_4 \\ 1, & otherwise \end{cases} \tag{11}$$

$$trf4\ (\beta_i^j) = atanh(\beta_i^j) = \frac{2}{\pi} \left|\arctan\left(\frac{\pi}{2} x\right)\right| \tag{12}$$

In equations, (5)-(12), *trf1, trf2, trf3, and trf4* can be defined as the transformation function applied on β_i^j and θ_1, θ_2, θ_3, θ_4 are the random numbers distributed over 0 and 1. The pseudo code of the proposed binary versions of MVO is shown in Algorithm 1 in which the position updating phase is modified using four different transfer functions.

Algorithm 1. Pseudo code of Binary version of Multiverse Optimization algorithm (BMVO)

```
Creation of U, where U is random universe.
Initalization of TDR, Best_universe and WEP.
SU=Sorted universes;  NF=Normalized inflation rate.
while end criterion is not met
Evaluate all the universes for the fitness or inflation rate
for each universe indexed by i
Update WEP and TDR ; Black_hole_index = i;
  for each object  indexed by j
      r1=random([0,1]);
    if r1<NF(Ui)
      White_hole_index= RouletteWheelSelection(-NF);
      U(Black_hole_index,j)=SU(White_hole_index,j);
    end if
      r2=random([0,1]);
    if r2<Wormhole_existance_probability
      r3= random([0,1]);
      r4= random([0,1]);
      if  r3<0.5
        Ω(t)=trf(Best_universe(j) + Travelling_distance_rate
        * ((ub(j) - lb(j)) * r4 + lb(j)));
        If  Ω(t)> random([0,1])
          Ω(t)=0;
        else
          Ω(t)=1;
        End if
          U(i ,j)=Ω(t)
      else
        Ω(t)=trf(Best_universe(j) - Travelling_distance_rate *
        ((ub(j) - lb(j)) * r4 + lb(j)));
        If  Ω(t)> random([0,1])
          Ω(t)=0;
        else
          Ω(t)=1;
        End if
          U(i,j)=Ω(t)
      end if
    end if
  end for
end for
end while
```

V. Materials and Methods

Generally, the problem of feature subset selection is assumed to be as a multi-objective optimization problem in which the occurrence of tradeoffs among two or more conflicting objectives is considered for making an optimal choice [50]. As discussed in the previous sections, the problem of feature subset selection consists of two most important objectives viz. to maximize the classification performance and to minimize the number of features selected or size of selected feature subset. The process feature selection is considered successful or solution is considered as the best solution if the dimensionality of the data is reduced and the accuracy of the algorithm improves or remains the same [51].

For an optimization algorithm, to weigh up the discriminative potential of each subset of features a fitness function must be defined. The fitness function for which every candidate solution of the

population is evaluated is based on k-NN classifier [52] as shown in equation (13).

$$\text{Fitness} = W_1 \left|\frac{FS}{N}\right| + W_2 * \mu(D) \quad (13)$$

Where $\mu(D)$ represents the error rate in the classification process calculated from the k-NN classifier. Furthermore, $|FS|$ is the total number of features selected by an optimization algorithm in the subset and $|N|$ is the original size(Number of features present) of feature set, W_1 and W_2 are two constants equivalent to the significance of classification eminence and feature subset size, where $W_1 \in [0, 1]$ and $W_2 = (1 - W_1)$ adopted from [16][53].

A. k-Nearest Neighbors

In this study, the simplest algorithm known as k-nearest neighbor's (k-NN) serves as an assessor of the candidate solutions in the population [70]. This algorithm is a supervised machine learning algorithm and also a non-parametric method used for classification as well as regression [52], which is based on finding k-nearest neighbors by means of smallest distance between training examples and the query instance [36].

B. Parameter Settings and Dataset Description

To summarize the system configuration for performing the experiments a CPU with a processor configuration of Core 2 Duo 2.00 GHz, 3.00 GB RAM and 64-bit Operating system is selected. The parameter setting for the proposed BMVO approaches is shown in Table II. Also, the algorithms used for comparison viz. GWO, SCA, WOA, and MVO are implemented for 10 different runs with 100 iterations in each run and are initialized to 20 agents. To carry out the experiments, 21 datasets were obtained from various online sources [54][55] which are summarized in Table III.

TABLE II. Various Parameter Settings for the Experiments

Parameter	Value
Runs	10
Number of Iterations	100
MIN_{WEP}	0.2
MAX_{WEP}	1
P(default value)	6
No. of Universes	20
Lb(Lower Bound)	0
Ub(Upper Bound)	1
Dimensions	Total features in the dataset
W_2(fitness function)	0.99
W_1(fitness function)	0.01

All the chosen datasets have dissimilar numbers of attributes plus instances representing a range of categories including medical domain also. The instances are arbitrarily divided into three different sets that are training, validation, and testing sets in a cross-validation manner for all data sets. The method for feature selection is based on k-NN classifier and works on trial and error logic. To limit the experiments, the value of k is set to 5 on all the datasets for its best output.

C. Evaluation Metrics

To quantify the performance of proposed algorithms and to compare with other existing metaheuristic algorithms, the following criteria are used which are summarized in Table IV along with their expected values:

a) **Average Classification Accuracy**- It can be described as the proportion of samples taken for testing correctly classified by the algorithm. It generally evaluates the ability of a classifier in classifying the dataset in N runs, where Acc_t is the accuracy obtained in t^{th} run.

b) **Average fitness** - This metric gives the average of the various fitness values achieved by a probabilistic algorithm in N runs, where Fit_t is the fitness obtained in t^{th} run.

c) **Worst fitness** - This metric gives a maximum of the N fitness values gained by the algorithm in N runs.

d) **Best fitness** - This measure gives a minimum of the N fitness values achieved by the algorithm in N runs.

e) **Standard Deviation** - This metric is defined as the divergence of the finest achieved solutions found after running a stochastic optimizer for N runs [36].

f) **Average Number of Features Selected** - This criterion is defined as the number of features selected averaged on all the runs.

g) **F-Measure** - F-measure is another important criterion to measure the performance of the classification algorithm also known as F-score; it is an assessment of classifier's accuracy, which integrates both the precision as well as the recall as a harmonic mean.

h) **Average Time** - This criterion measures the time taken by an algorithm averaged on all the runs.

i) **Non-Parametric Testing** - Wilcoxon's signed rank test is one of the non-parametric tests that aims to detect significant differences between the two sample means. The test returns a parameter called p-value which verifies the level of significance of two algorithms [56].

TABLE III. Datasets Description with No. of Instances and Attributes

No.	Dataset	Instances	Attributes
D_1	Zoo	101	16
D_2	Statlog	1000	20
D_3	Lung cancer	32	56
D_4	Exactly	1000	13
D_5	Exactly2	1000	13
D_6	M-of-N	1000	13
D_7	Heart	294	13
D_8	Vote	300	16
D_9	Spect Heart	267	22
D_{10}	Australian	690	14
D_{11}	Ionosphere	351	34
D_{12}	Water treatment	521	38
D_{13}	Wine	178	13
D_{14}	Indian Liver	583	10
D_{15}	Tic-Tac-Toe	958	9
D_{16}	Wavform	1000	21
D_{17}	Dermatology	366	34
D_{18}	Glass Identification	214	9
D_{19}	Breast cancer	699	9
D_{20}	Sonar	208	60
D_{21}	Vowel	990	13

TABLE IV. EVALUATION METRICS

Evaluation criteria	Formula	Expected Value
Average Classification Accuracy	$Avg\_Acc = \frac{1}{N}\sum_{t=1}^{N} Acc_t$	Maximum
Average fitness	$Avg\_fit = \frac{1}{N}\sum_{t=1}^{N} Fit_t$	Minimum
Worst fitness	$Worst\_fit = \max(Fit_1 : Fit_N)$	Maximum
Best fitness	$Best\_fit = \min(Fit_1 : Fit_N)$	Minimum
Standard Deviation	$St\_dev = \sqrt{\frac{\sum_{t=1}^{N}(Fit_t - \overline{Fit})^2}{N-1}}$	Minimum
Average size of Feature Subset	$Avg\_FS = \frac{1}{N}\sum_{k=1}^{N} \text{SizeFS}_t$	Minimum
F-Measure	$F\_Measure = \frac{2.\text{Precision.Recall}}{(Precision+Recall)}$	Maximum
Average Time	$Avg\_Time = \frac{1}{N}\sum_{t=1}^{N} \text{time}_t$	Minimum

VI. RESULTS AND DISCUSSIONS

This section reports all the results obtained by implementing different variants of Binary Multi-verse optimization (BMVO). Also, the proposed binary MVO approaches are compared with other binary versions of metaheuristic algorithms viz. GWO, WOA, SCA, PSO, ALO, and bMVO(without transformation functions) based on the criteria shown in Table IV. It can be depicted from Table V that when algorithms are compared on the basis of average classification accuracy BMVO-I and BMVO-II outperforms the other versions of BMVO as well as another state of the art algorithms considered for comparison. .Moving ahead, another important metric to assess the performance of the proposed variants of BMVO is the average fitness value obtained from the objective function on various datasets. Clearly, it can be depicted from Table VI that all the versions of BMVO outperform the binary versions of existing meta-heuristic algorithms viz. GWO, WOA, SCA, PSO, ALO bMVO(native version, without transformation function) on almost all the datasets but BMVO-I and BMVO-II perform better than other two versions i.e. BMVO-III and BMVO-IV in terms of average fitness values obtained.

Similarly Table VII and Table VIII compares the performance of proposed versions with the other state of the art metaheuristic algorithms on the basis of their worst (Maximum in this case) and best(Minimum in this case) fitness values, obtained in various runs on various datasets, On comparison it can be found that all the four versions outperform the other metaheuristic algorithms in terms of worst fitness. However, the performance of BMVO-I and BMVO-II is better than BMVO-III and BMVO-IV and GWO, SCA, WOA and bMVO in terms of worst fitness obtained, consequently in terms of best fitness value obtained, all the versions of BMVO obtain Best results on most of the datasets taken for comparison as shown in Table VIII.

Moving ahead Table IX shows the performance comparison of all the algorithms on the basis of standard deviation achieved. Clearly, it can be seen that proposed versions of BMVO perform better than GWO, WOA, SCA, PSO, ALO, bMVO on almost 14 different datasets in terms of standard deviation out of remaining datasets MVO performs better on three different datasets and GWO, WOA, SCA performs better in terms of a standard deviation on 5 different datasets.

Moving further another important measure to access the performance of the proposed versions of BMVO is the average number of features which are selected for feature subset by various algorithms. Clearly it can be deduced from Table X that proposed variants outperforms GWO, WOA, SCA, PSO, ALO, bMVO in terms of average numbers of selected features, also it can be noticed that out of four different variants of BMVO, the proposed BMVO-I and BMVO-II outperforms the other two variants of BMVO in terms of average number of selected features. Also, the graph shown in Fig. 2 presents a number of total features selected on all datasets by all algorithms. It can be clearly depicted from the graph that the proposed versions of BMVO outperform all the existing metaheuristic algorithms when the comparison was made on the basis of an average number of features selected on all dataset. Moving ahead, another important criterion to gauge the performance of proposed variants is F-Measure, which makes use of precision and recall parameter. It can be depicted from the graph in Fig. 3 that proposed versions of BMVO outperform the other algorithms when the comparison was made on the basis of average F-measure values obtained on all the datasets. The results of F-measure values can be seen in Table XV in the appendix section, that indicate that the proposed algorithms perform much better on the maximum number of datasets.

Similarly, when the comparison is made on the basis of average time taken by an algorithm per run in as shown in Table XIV in the appendix section, the proposed variants BMVO-II and BMVO-III show much better performance than other algorithms. Also it can be clearly seen from the graph shown in Fig. 4 that all the binary variants (apart from BMVO-I) of MVO outperform another state of the art metaheuristic algorithms but BMVO-II and BMVO-III takes lesser time when cumulative sum of average time per run is taken on all the datasets than all the other variants of BMVO as well as the metaheuristic algorithms considered for comparison. The overall performance of BMVO-I is better than the remaining algorithms in terms of an average number of features selected on nine datasets and average classification accuracy on eleven datasets. The use of transformation function leads to an increase in the execution time of the algorithm to some extent.

It can be clearly depicted from Table XI which indicates the p-values obtained, in most of the cases the results are significant at $p \leq 0.05$. To summarize the results it can be clearly seen that that when comparison is made on the basis of average classification accuracy and average fitness value BMVO-I outperforms all the other versions of BMVO: *BMVO-I > BMVO-II > BMVO-III > BMVO-IV*, on the other hand when comparison was made on the basis of standard deviation BMVO-III outperforms all the other versions of BMVO: *BMVO-II > BMVO-III > BMVO-I >BMVO-IV*. However when the comparison was made on the basis of the average number of selected features and f- measure. BMVO-I outperforms all the other variants of BMVO: *BMVO-I > BMVO-II > BMVO-III > BMVO-IV* and finally when the comparison is made on the basis of average time taken by an algorithm per run BMVO-III performs better than other algorithms: *BMVO-III > BMVO-II > BMVO-IV > BMVO-I.*

VII. CONCLUSIONS AND FUTURE WORK

In recent times, diverse metaheuristic algorithms have been developed to find the solution for feature subset selection problem in different applications. In this research work, four different wrapper based binary variants of Multi-Verse Optimization algorithm is proposed by converting the continuous version of Multi-verse

TABLE V. COMPARATIVE ANALYSIS OF AVERAGE CLASSIFICATION ACCURACY OBTAINED

	GWO	*WOA*	*SCA*	*PSO*	*ALO*	*bMVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	0.919	0.910	0.905	0.943	0.884	0.939	0.937	**0.961**	0.953	0.935
D_2	0.717	0.718	0.716	0.710	0.713	0.721	**0.743**	0.740	0.730	0.727
D_3	0.863	0.888	0.881	0.894	0.925	0.888	**0.956**	0.900	0.938	0.875
D_4	0.703	0.726	0.718	0.723	0.704	0.738	0.879	**0.974**	0.791	0.808
D_5	0.754	0.748	0.748	0.755	0.760	0.752	**0.768**	0.763	0.765	0.765
D_6	0.880	0.911	0.831	0.900	0.881	0.912	0.970	**0.977**	0.950	0.937
D_7	0.805	0.821	0.816	0.817	0.816	0.834	**0.854**	0.846	0.829	0.831
D_8	0.943	0.943	0.939	0.931	0.946	0.954	**0.962**	0.957	0.954	0.959
D_9	0.719	0.726	0.737	0.729	0.728	0.754	**0.774**	0.757	0.757	0.754
D_{10}	0.799	0.777	0.790	0.810	0.790	0.820	**0.870**	0.860	0.849	0.844
D_{11}	0.866	0.867	0.870	0.854	0.869	0.861	**0.899**	0.888	0.897	0.863
D_{12}	0.815	0.821	0.815	0.807	0.821	0.828	**0.855**	**0.855**	0.841	0.843
D_{13}	0.974	0.980	0.978	0.979	0.983	0.984	**0.994**	0.991	0.990	0.991
D_{14}	0.703	0.717	0.710	0.704	0.705	0.717	0.723	**0.734**	0.729	0.728
D_{15}	0.792	0.794	0.756	**0.808**	0.798	0.797	0.790	0.798	0.801	0.799
D_{16}	0.823	0.832	0.801	0.830	0.825	**0.844**	0.833	0.836	0.830	0.830
D_{17}	0.962	0.975	0.948	0.974	0.967	**0.985**	0.972	0.983	0.977	0.981
D_{18}	0.664	0.674	0.682	0.680	0.674	0.686	0.687	0.699	**0.700**	0.698
D_{19}	0.971	0.969	0.965	0.975	0.973	0.973	0.977	0.976	**0.978**	0.975
D_{20}	0.783	0.767	0.772	0.785	0.776	0.794	**0.840**	0.834	0.837	0.825
D_{21}	0.820	0.836	0.790	0.823	0.812	0.839	0.842	0.848	**0.852**	0.835

TABLE VI. COMPARATIVE ANALYSIS OF AVERAGE FITNESS VALUE OBTAINED

	GWO	*WOA*	*SCA*	*PSO*	*ALO*	*bMVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	0.162	0.165	0.144	0.101	0.218	0.140	0.101	**0.063**	0.101	0.103
D_2	0.305	0.308	0.297	0.303	0.302	0.299	0.284	**0.279**	0.289	0.293
D_3	0.251	0.192	0.252	0.190	0.129	0.192	**0.128**	0.250	0.251	0.253
D_4	0.328	0.307	0.335	0.305	0.333	0.296	0.271	**0.091**	0.287	0.288
D_5	0.269	0.267	0.271	0.264	0.273	0.258	0.255	**0.252**	0.255	**0.252**
D_6	0.157	0.134	0.140	0.131	0.153	0.137	0.124	**0.077**	0.113	0.103
D_7	0.242	0.225	0.238	0.220	0.219	0.200	**0.192**	0.197	0.195	0.225
D_8	0.085	0.099	0.081	0.112	0.076	0.078	0.068	0.065	0.066	**0.058**
D_9	0.329	0.308	0.307	0.329	0.316	0.296	**0.262**	0.276	0.271	0.278
D_{10}	0.273	0.277	0.265	0.273	0.264	0.265	**0.146**	0.160	0.205	0.181
D_{11}	0.173	0.175	0.169	0.199	0.155	0.186	0.138	**0.137**	0.151	0.184
D_{12}	0.239	0.241	0.213	0.227	0.202	0.203	0.175	**0.171**	0.191	0.210
D_{13}	0.053	0.040	0.031	0.041	0.050	0.030	**0.017**	0.037	0.018	0.027
D_{14}	0.316	0.300	0.309	0.328	0.324	0.308	0.294	**0.290**	**0.290**	0.292
D_{15}	0.229	0.244	0.230	0.217	0.227	0.231	0.236	**0.224**	0.226	0.230
D_{16}	0.206	0.193	0.191	0.189	0.196	**0.175**	0.182	0.186	0.191	0.183
D_{17}	0.070	0.047	0.057	0.050	0.053	0.030	0.043	**0.027**	0.043	0.033
D_{18}	0.403	0.385	0.377	0.368	0.394	0.376	0.368	0.357	0.357	**0.354**
D_{19}	0.053	0.052	0.043	0.036	0.042	0.043	**0.034**	0.037	0.037	0.038
D_{20}	0.299	0.290	0.252	0.274	0.252	0.309	**0.195**	0.213	0.224	0.252
D_{21}	0.229	0.209	0.228	0.200	0.233	0.220	0.203	0.191	**0.175**	0.200

TABLE VII. COMPARATIVE ANALYSIS OF WORST FITNESS VALUE OBTAINED

	GWO	*WOA*	*SCA*	*PSO*	*ALO*	*bMVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	0.043	0.044	0.046	0.028	0.046	0.004	0.003	0.024	**0.005**	0.025
D_2	0.258	0.273	0.263	0.281	0.265	0.258	**0.234**	0.248	0.262	0.244
D_3	0.005	0.067	0.007	0.066	0.003	0.067	**0.003**	**0.003**	**0.003**	0.005
D_4	0.277	0.237	0.071	0.240	0.284	0.202	0.011	**0.005**	0.071	**0.005**
D_5	0.225	0.246	0.221	0.221	0.221	0.241	**0.205**	0.217	0.209	0.219
D_6	0.087	**0.005**	0.048	0.068	0.073	0.054	**0.005**	**0.005**	**0.005**	**0.005**
D_7	0.150	0.141	0.132	0.153	0.158	0.131	0.125	**0.117**	0.153	0.137
D_8	0.050	0.044	0.046	0.055	0.043	0.040	**0.016**	0.035	0.030	0.033
D_9	0.242	0.250	0.227	0.228	0.234	0.231	**0.189**	0.205	0.211	0.204
D_{10}	0.142	0.150	0.141	0.158	0.165	0.138	**0.107**	0.126	0.123	0.131
D_{11}	0.106	0.101	0.082	0.118	0.121	0.090	0.081	0.083	**0.073**	0.101
D_{12}	0.147	0.133	0.144	0.129	0.153	0.129	0.119	**0.113**	0.130	0.131
D_{13}	0.006	0.017	0.006	0.008	0.007	0.017	**0.003**	**0.003**	0.005	0.004
D_{14}	0.285	0.257	0.273	0.265	0.272	0.259	0.252	**0.230**	0.253	0.258
D_{15}	0.196	0.190	**0.182**	0.177	0.190	0.190	0.188	0.183	0.187	0.184
D_{16}	0.157	0.160	0.154	0.165	0.162	**0.147**	0.157	0.158	0.159	0.160
D_{17}	0.021	0.014	**0.013**	0.015	0.026	0.017	0.020	0.015	0.020	0.015
D_{18}	0.272	0.290	0.264	0.278	0.265	0.256	0.273	**0.246**	0.272	0.255
D_{19}	0.025	0.024	0.028	0.024	0.024	**0.020**	0.023	**0.020**	**0.020**	0.021
D_{20}	0.147	0.205	0.185	0.169	0.205	0.158	0.128	0.139	0.137	**0.109**
D_{21}	**0.129**	0.144	0.157	0.169	0.163	0.130	0.136	0.138	0.134	0.151

TABLE VIII. COMPARATIVE ANALYSIS OF BEST FITNESS VALUE OBTAINED

	GWO	*WOA*	*SCA*	*PSO*	*ALO*	*bMVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	0.043	0.044	0.046	0.028	0.046	0.004	0.003	0.024	**0.005**	0.025
D_2	0.258	0.273	0.263	0.281	0.265	0.258	**0.234**	0.248	0.262	0.244
D_3	0.005	0.067	0.007	0.066	0.003	0.067	**0.003**	**0.003**	**0.003**	0.005
D_4	0.277	0.237	0.071	0.240	0.284	0.202	0.011	**0.005**	0.071	**0.005**
D_5	0.225	0.246	0.221	0.221	0.221	0.241	**0.205**	0.217	0.209	0.219
D_6	0.087	**0.005**	0.048	0.068	0.073	0.054	**0.005**	**0.005**	**0.005**	**0.005**
D_7	0.150	0.141	0.132	0.153	0.158	0.131	0.125	**0.117**	0.153	0.137
D_8	0.050	0.044	0.046	0.055	0.043	0.040	**0.016**	0.035	0.030	0.033
D_9	0.242	0.250	0.227	0.228	0.234	0.231	**0.189**	0.205	0.211	0.204
D_{10}	0.142	0.150	0.141	0.158	0.165	0.138	**0.107**	0.126	0.123	0.131
D_{11}	0.106	0.101	0.082	0.118	0.121	0.090	0.081	0.083	**0.073**	0.101
D_{12}	0.147	0.133	0.144	0.129	0.153	0.129	0.119	**0.113**	0.130	0.131
D_{13}	0.006	0.017	0.006	0.008	0.007	0.017	**0.003**	**0.003**	0.005	0.004
D_{14}	0.285	0.257	0.273	0.265	0.272	0.259	0.252	**0.230**	0.253	0.258
D_{15}	0.196	0.190	**0.182**	0.177	0.190	0.190	0.188	0.183	0.187	0.184
D_{16}	0.157	0.160	0.154	0.165	0.162	**0.147**	0.157	0.158	0.159	0.160
D_{17}	0.021	0.014	**0.013**	0.015	0.026	0.017	0.020	0.015	0.020	0.015
D_{18}	0.272	0.290	0.264	0.278	0.265	0.256	0.273	**0.246**	0.272	0.255
D_{19}	0.025	0.024	0.028	0.024	0.024	**0.020**	0.023	**0.020**	**0.020**	0.021
D_{20}	0.147	0.205	0.185	0.169	0.205	0.158	0.128	0.139	0.137	**0.109**
D_{21}	**0.129**	0.144	0.157	0.169	0.163	0.130	0.136	0.138	0.134	0.151

TABLE IX. Comparative Analysis of Standard Deviation Value

	GWO	*WOA*	*SCA*	*PSO*	*ALO*	*bMVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	0.037	0.043	0.030	0.022	0.049	0.041	0.036	**0.016**	0.030	0.030
D_2	0.012	0.010	0.011	0.007	0.011	0.013	0.014	0.010	**0.009**	0.016
D_3	0.076	0.057	0.063	0.041	0.049	**0.039**	0.059	0.093	0.077	0.072
D_4	**0.015**	0.027	0.095	0.020	0.018	0.030	0.093	0.033	0.088	0.099
D_5	0.014	0.007	0.013	0.012	0.015	**0.006**	0.017	0.013	0.015	0.010
D_6	0.024	0.036	0.026	0.019	0.025	0.025	0.040	**0.022**	0.046	0.033
D_7	0.033	0.026	0.029	0.022	0.022	0.022	0.023	0.026	**0.013**	0.029
D_8	0.011	0.016	**0.009**	0.017	0.011	0.011	0.015	**0.009**	0.013	**0.009**
D_9	0.029	0.020	0.024	0.028	0.026	**0.017**	0.019	0.020	0.022	0.020
D_{10}	0.049	0.037	0.042	0.034	0.032	0.036	0.012	**0.011**	0.024	0.015
D_{11}	0.021	0.027	0.029	0.026	0.010	0.024	0.020	**0.016**	0.025	0.022
D_{12}	0.028	0.034	0.026	0.027	0.014	0.022	0.019	**0.016**	0.020	0.025
D_{13}	0.015	0.009	0.008	0.011	0.013	0.006	0.006	0.010	**0.004**	0.009
D_{14}	0.011	0.013	**0.010**	0.018	0.015	0.015	0.015	0.019	**0.010**	0.012
D_{15}	**0.008**	0.019	0.014	0.012	0.012	0.016	0.013	0.014	0.012	0.015
D_{16}	0.016	0.013	0.012	0.009	0.012	0.010	0.008	0.009	0.010	**0.007**
D_{17}	0.014	0.010	0.014	0.011	0.009	0.005	0.008	**0.004**	0.008	0.006
D_{18}	0.047	0.034	0.037	0.032	0.041	0.036	**0.025**	0.036	0.032	0.029
D_{19}	0.009	0.007	0.005	0.004	0.006	0.008	**0.003**	0.006	0.006	0.006
D_{20}	0.050	0.024	**0.020**	0.031	0.014	0.045	0.024	**0.020**	0.025	0.040
D_{21}	0.029	0.018	0.021	0.011	0.020	0.024	0.020	0.017	**0.013**	0.016

TABLE X. Comparative Analysis of No. of Features Selected

	GWO	*WOA*	*SCA*	*PSO*	*ALO*	*bMVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	10.3	11.3	11.3	10.5	9.4	11.1	**6.7**	7.8	8.3	8.3
D_2	11.3	11.7	10	12	9.8	12.8	**6.8**	7	9.3	8.5
D_3	24	32.4	27.2	27	24.8	33.4	18.6	**17**	23	26.6
D_4	10	10.3	8.8	9.9	10	9.6	7.1	**6.3**	7.9	7.3
D_5	6.7	7.8	5.1	6	8.9	5.6	3.5	**2.7**	3.6	4.9
D_6	10.9	10.1	10.1	10.2	11.8	9.7	6.8	**6.6**	7.6	8
D_7	5.2	5.6	6.4	5.8	4.2	7.7	4.1	**3.8**	4.9	4.2
D_8	8.7	7.6	9	8.1	8.6	9.6	**4.5**	6.6	7.2	8
D_9	11.1	14.2	12.8	12.4	11.8	15.9	8.9	**8.5**	11.2	8.8
D_{10}	6	6.8	6.9	6.2	6.1	6.5	**3.7**	4.2	4.7	5.2
D_{11}	13.9	20.2	19	18.9	16.4	21	**9**	9.7	16.8	13.4
D_{12}	19	22.6	21.4	20.7	17.2	25.9	**14.4**	15.2	18.4	17.8
D_{13}	7.4	8.3	9.3	9.1	8.2	8.2	5.8	**5.5**	6.8	6.5
D_{14}	3.8	4.9	5	4.3	5.1	5.1	**2.3**	3.1	2.6	4.5
D_{15}	9	8.8	8.7	8.4	9	8	**5.8**	5.9	7.1	7.2
D_{16}	19.5	18.8	17.5	19	19.4	17.8	13.7	**13.3**	13.9	13.6
D_{17}	26.6	26.1	26.2	26.4	27.9	26.8	**14.5**	16.1	20.2	18.7
D_{18}	4.9	6.3	5.7	6.3	5.5	5.8	**4.3**	4.7	**4.3**	4.5
D_{19}	6	6.7	6.2	5.5	4.8	5.8	4.8	5	**4.6**	4.8
D_{20}	32.5	28.1	32.7	31	28.9	42.9	**22**	23.9	30	26.9
D_{21}	9.4	9.5	8.7	9.5	8.7	9.6	**7.7**	**7.7**	8.2	8.3

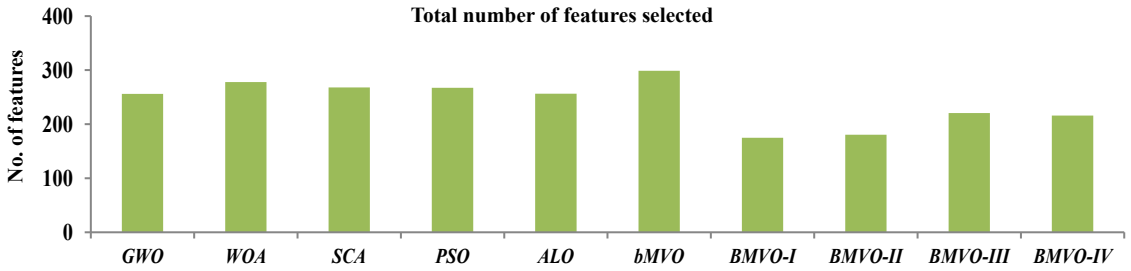


Fig 2. Graphical representation of No. of Features selected Averaged on all datasets by all algorithms.

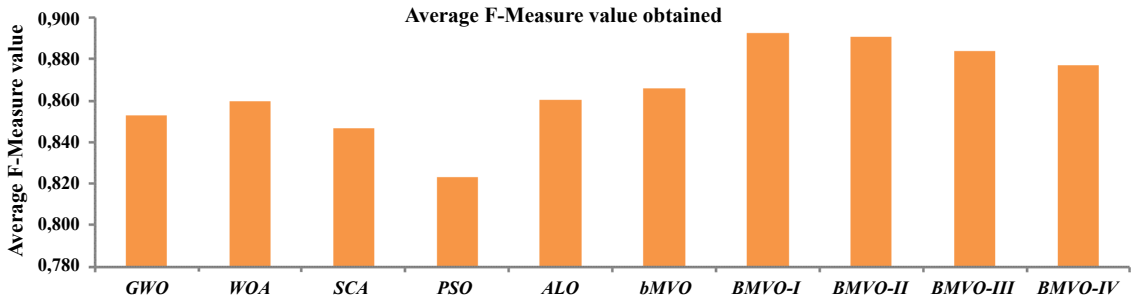


Fig 3. Graphical representation of average of cumulative sum of F-Measure values (average F-measure values on all the datasets).

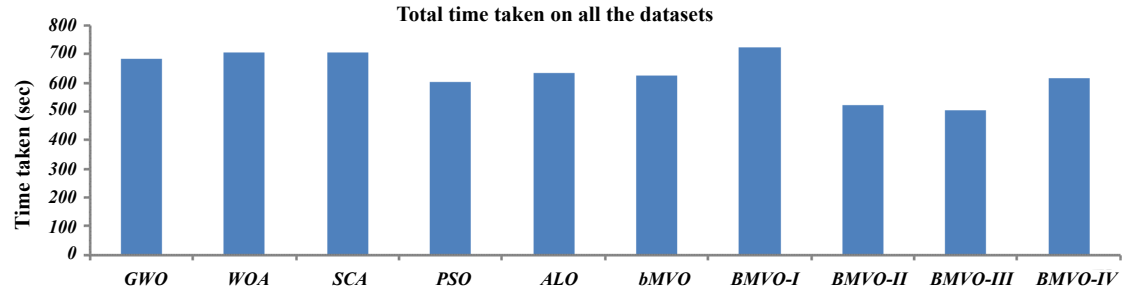


Fig 4. Graphical representation of cumulative sum of average time taken per run by an algorithm on all the datasets.

TABLE XI. Wilcoxon Signed Rank Test for the Average Fitness Obtained by the Different Algorithms

	GWO	WOA	SCA	PSO	ALO	bMVO	BMVO-I	BMVO-II	BMVO-III	BMVO-IV
bMVO	0.00014	0.00034	0.01352	0.00124	0.00614	-	0.0009	0.00012	0.00058	0.01016
BMVO-I	0	0.0001	0.00014	0.00038	0.0001	0.0009	-	**0.8493**	0.04236	0.00374
BMVO-II	0	0	8.00E-05	0.00014	0.00032	0.00012	**0.8493**	-	0.04338	0.00016
BMVO-III	0	8.00E-05	8.00E-05	0.00014	0.00001	0.00058	0.04236	0.04338	-	0.0466
BMVO-IV	6.00E-05	0.00058	0.00116	0.00158	.00116	0.01016	0.00374	0.00016	0.0466	-

optimization algorithm into its binary variants and are applied on one of the important problems of machine learning, i.e. feature selection. These binary variants make use of different types of transformation functions from the class of V-shaped as well as S-shaped transformation functions, which defines the likelihood of altering a position vector's elements from 0 to 1 and vice versa. Their performance is compared against the six most recent metaheuristic algorithms used for feature selection viz. Grey wolf Optimization (GWO), Whale Optimization Algorithm (WOA), Sine Cosine Optimization algorithm (SCA), Particle Swarm Optimization (PSO) Ant lion Optimization (ALO), and Multi-verse Optimization Algorithm (MVO) and their performance are analyzed on the basis of various evaluation criteria. To cope with the stochastic nature of the metaheuristic algorithms the results from all the algorithms were averaged on ten different runs. The obtained evaluation results clearly indicate that the proposed variants of BMVO outperform the existing metaheuristic to find the solution of feature subset selection problem. Furthermore, it can be seen that the performance of all the variants of BMVO is better than algorithms taken for comparison but the performance of BMVO versions based on s-shaped(sigmoid and modified sigmoid) transformation function is better than the performance of variants based on V-shaped(tangent hyperbolic and inverse tangent) transformation function.

Furthermore, the MVO makes use of selection mechanism which is run for every variable in every universe over the iterations also the use of sorting algorithm in every iteration, along with these two basic operations the binary MVO makes use of transfer functions for the

conversion of continuous version of MVO to binary version which may lead to increase in the time complexity of the algorithm as compared to other binary algorithms considered for the comparison which can be gauged from the graph shown in Fig. 4.

In the future, the proposed binary version of the Multi-verse optimization algorithm can be used to explore for the solutions of other binary optimization problems, which may lead to an improvement in the results. Moreover, this work can be enhanced further by developing other different variants of MVO based on other transformation functions or combining the MVO algorithms with local search approaches like simulated annealing and tabu search for feature selection problem. Lastly, the hybrid versions of MVO can be developed by hybridizing MVO with some other metaheuristic algorithms.

Appendix

Similarly Table XII and Table XIII shows the detailed comparative analysis of maximum and minimum classification accuracies obtained by different versions of BMVO with GWO, WOA, SCA and original MVO (without incorporating transformation functions) for feature selection in all the 10 runs, this measure gives an idea of variation of outcome in all the runs.

1. **Maximum Accuracy**- This criterion gives the highest percentage of test samples correctly classified among all the runs, where the function max calculates the maximum accuracy from all runs.
2. **Minimum Accuracy**- This measure gives the lowest percentage of test samples correctly classified among all N the runs, where the function min calculates the minimum accuracy from all runs.

TABLE XII. Comparative Analysis of maximum Classification Accuracy Obtained

MAX	*GWO*	*WOA*	*SCA*	*PSO*	*ALO*	*MVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	0.961	0.961	0.941	0.980	0.961	1.000	1.000	0.980	1.000	0.980
D_2	0.748	0.732	0.730	0.724	0.736	0.744	0.768	0.752	0.740	0.758
D_3	1.000	0.938	0.938	0.938	1.000	0.938	1.000	1.000	1.000	1.000
D_4	0.730	0.768	0.894	0.764	0.722	0.802	0.994	1.000	0.934	1.000
D_5	0.782	0.762	0.778	0.780	0.780	0.762	0.794	0.782	0.790	0.784
D_6	0.918	1.000	0.864	0.938	0.932	0.952	1.000	1.000	1.000	1.000
D_7	0.850	0.864	0.871	0.850	0.844	0.871	0.878	0.884	0.850	0.864
D_8	0.953	0.960	0.947	0.953	0.960	0.967	0.987	0.967	0.973	0.973
D_9	0.761	0.754	0.776	0.776	0.769	0.776	0.813	0.799	0.791	0.799
D_{10}	0.861	0.852	0.861	0.843	0.838	0.864	0.896	0.875	0.878	0.872
D_{11}	0.898	0.903	0.920	0.886	0.881	0.915	0.920	0.921	0.932	0.903
D_{12}	0.858	0.874	0.858	0.874	0.851	0.877	0.885	0.889	0.874	0.874
D_{13}	1.000	0.989	0.989	1.000	1.000	0.989	1.000	1.000	1.000	1.000
D_{14}	0.716	0.747	0.723	0.736	0.729	0.740	0.747	0.771	0.747	0.747
D_{15}	0.812	0.816	0.785	0.831	0.818	0.818	0.820	0.820	0.820	0.825
D_{16}	0.852	0.848	0.832	0.842	0.846	0.860	0.848	0.848	0.846	0.844
D_{17}	0.984	0.995	0.967	0.995	0.984	0.989	0.984	0.989	0.984	0.989
D_{18}	0.729	0.710	0.738	0.729	0.738	0.748	0.729	0.757	0.729	0.748
D_{19}	0.980	0.980	0.974	0.983	0.983	0.986	0.983	0.986	0.986	0.983
D_{20}	0.856	0.798	0.817	0.837	0.798	0.846	0.875	0.865	0.865	0.894
D_{21}	0.875	0.861	0.820	0.836	0.840	0.877	0.869	0.867	0.871	0.855

TABLE XIII. Comparative Analysis of minimum Classification Accuracy Obtained

MIN	*GWO*	*WOA*	*SCA*	*PSO*	*ALO*	*MVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	0.843	0.843	0.863	0.902	0.784	0.863	0.902	0.941	0.902	0.902
D_2	0.696	0.698	0.704	0.698	0.702	0.706	0.716	0.722	0.712	0.708
D_3	0.750	0.813	0.750	0.813	0.875	0.813	0.875	0.750	0.750	0.750
D_4	0.672	0.698	0.664	0.700	0.674	0.710	0.732	0.914	0.716	0.712
D_5	0.734	0.734	0.730	0.740	0.730	0.742	0.746	0.750	0.746	0.752
D_6	0.852	0.874	0.788	0.876	0.856	0.870	0.882	0.928	0.894	0.904
D_7	0.762	0.776	0.762	0.782	0.782	0.803	0.810	0.803	0.810	0.776
D_8	0.920	0.907	0.920	0.893	0.927	0.927	0.933	0.940	0.940	0.947
D_9	0.672	0.694	0.694	0.672	0.687	0.709	0.739	0.724	0.731	0.724
D_{10}	0.730	0.725	0.725	0.730	0.739	0.739	0.855	0.841	0.797	0.820
D_{11}	0.830	0.830	0.835	0.807	0.847	0.818	0.864	0.864	0.852	0.818
D_{12}	0.766	0.762	0.782	0.774	0.801	0.801	0.828	0.831	0.812	0.793
D_{13}	0.955	0.966	0.966	0.966	0.955	0.978	0.989	0.966	0.989	0.978
D_{14}	0.685	0.702	0.695	0.671	0.678	0.695	0.705	0.712	0.712	0.709
D_{15}	0.779	0.764	0.733	0.789	0.781	0.777	0.766	0.781	0.777	0.777
D_{16}	0.802	0.814	0.776	0.818	0.812	0.832	0.822	0.818	0.814	0.822
D_{17}	0.934	0.962	0.923	0.956	0.956	0.978	0.962	0.978	0.962	0.973
D_{18}	0.598	0.617	0.626	0.636	0.607	0.626	0.636	0.645	0.645	0.645
D_{19}	0.954	0.954	0.957	0.969	0.963	0.963	0.971	0.969	0.969	0.966
D_{20}	0.702	0.712	0.750	0.731	0.750	0.692	0.808	0.788	0.779	0.750
D_{21}	0.776	0.798	0.756	0.806	0.774	0.786	0.802	0.812	0.830	0.804

TABLE XIV. COMPARATIVE ANALYSIS OF **AVERAGE TIME TAKEN PER RUN (100 ITERATIONS)** OBTAINED

	GWO	*WOA*	*SCA*	*PSO*	*ALO*	*bMVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	20.11	27.63	**18.04**	17.04	27.17	23.13	47.24	23.21	19.30	26.47
D_2	45.68	39.64	46.78	36.45	54.48	38.06	47.83	**26.02**	42.25	71.19
D_3	24.16	31.42	21.01	16.55	25.53	**17.27**	32.60	22.43	17.91	36.06
D_4	29.88	34.47	33.96	31.28	39.47	31.57	40.00	28.17	**26.43**	30.86
D_5	29.90	59.59	35.21	34.12	33.56	30.77	38.56	**27.64**	30.38	32.15
D_6	30.32	51.98	34.59	30.48	35.64	31.53	34.02	34.25	**27.88**	27.95
D_7	22.81	32.75	17.41	17.31	18.93	18.16	20.35	25.76	16.94	**16.83**
D_8	20.63	35.64	18.59	17.18	20.09	17.42	20.57	21.20	**17.21**	18.90
D_9	31.01	38.13	23.55	16.96	24.50	17.27	20.49	24.13	**16.78**	17.78
D_{10}	32.08	41.62	30.41	20.58	29.24	23.74	22.85	27.60	**20.19**	20.37
D_{11}	38.44	22.81	30.25	30.93	26.58	19.95	33.35	19.00	**18.80**	19.81
D_{12}	35.91	34.98	31.25	35.05	31.04	29.72	**20.05**	22.29	22.64	25.52
D_{13}	39.84	23.05	23.35	16.75	19.24	20.28	24.24	19.51	**18.13**	23.56
D_{14}	39.96	20.45	24.37	27.65	21.19	**20.05**	22.97	22.75	21.13	20.30
D_{15}	47.50	31.22	64.10	37.42	42.66	39.45	33.84	23.74	**23.58**	24.15
D_{16}	46.19	36.67	72.19	47.06	45.80	39.59	55.11	**28.17**	32.25	32.41
D_{17}	25.59	22.91	35.41	33.33	26.77	22.96	55.86	**22.29**	27.67	28.99
D_{18}	22.38	36.31	34.33	29.79	20.54	20.59	39.84	**20.34**	22.76	41.20
D_{19}	28.39	25.25	38.92	28.24	25.84	22.09	37.21	**18.65**	20.01	32.16
D_{20}	22.42	18.64	31.43	16.87	25.71	**17.69**	24.97	18.48	30.94	25.07
D_{21}	51.97	41.71	42.70	61.94	42.17	122.26	51.23	45.43	**31.66**	42.77

TABLE XV. COMPARATIVE ANALYSIS OF **F-MEASURE VALUES** OBTAINED

	GWO	*WOA*	*SCA*	*PSO*	*ALO*	*bMVO*	*BMVO-I*	*BMVO-II*	*BMVO-III*	*BMVO-IV*
D_1	0.980	**0.994**	0.959	0.963	0.965	0.978	0.973	0.993	0.993	0.985
D_2	0.814	0.812	0.811	0.795	0.807	0.812	**0.828**	0.825	0.819	0.817
D_3	0.705	0.746	0.721	0.552	0.821	0.747	**0.924**	0.796	0.861	0.765
D_4	0.800	0.813	0.813	0.795	0.806	0.820	0.913	**0.981**	0.854	0.867
D_5	0.856	0.851	0.853	0.837	0.857	0.855	**0.865**	**0.865**	**0.865**	0.864
D_6	0.837	0.876	0.763	0.836	0.836	0.882	0.958	**0.969**	0.933	0.914
D_7	0.849	0.864	0.859	0.809	0.860	0.874	**0.888**	0.880	0.870	0.873
D_8	0.929	0.927	0.926	0.892	0.935	0.942	**0.953**	0.946	0.945	0.949
D_9	0.764	0.774	0.785	0.748	0.777	0.793	**0.812**	0.802	0.802	0.799
D_{10}	0.822	0.802	0.815	0.731	0.816	0.839	**0.879**	0.870	0.859	0.857
D_{11}	0.904	0.904	0.907	0.883	0.906	0.901	0.926	0.919	**0.927**	0.900
D_{12}	0.891	0.897	0.892	0.859	0.895	0.900	**0.915**	0.912	0.906	0.907
D_{13}	0.980	0.986	0.980	0.979	0.989	0.986	**0.995**	**0.995**	0.991	0.991
D_{14}	0.806	0.816	0.809	0.786	0.805	0.814	0.825	**0.829**	0.825	0.820
D_{15}	0.856	0.857	0.827	0.861	0.862	0.859	0.848	0.854	**0.860**	0.858
D_{16}	0.780	0.789	0.745	0.779	0.780	**0.809**	0.787	0.791	0.781	0.784
D_{17}	0.996	0.999	0.990	0.953	0.995	0.999	0.993	**1.000**	0.997	0.998
D_{18}	0.698	0.698	0.725	0.678	0.721	0.713	0.735	0.740	**0.755**	0.751
D_{19}	0.978	0.976	0.973	0.976	0.979	0.979	0.982	0.982	**0.983**	0.981
D_{20}	0.743	0.728	0.734	0.678	0.739	0.753	**0.819**	0.808	0.809	0.794
D_{21}	0.918	0.945	0.892	0.903	0.912	0.933	0.929	0.942	0.927	**0.949**

References

[1] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), 16-28.

[2] Hancer, E., Xue, B., Karaboga, D., & Zhang, M. (2015). A binary ABC algorithm based on advanced similarity scheme for feature selection. *Applied Soft Computing*, 36, 334-348.

[3] Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2017). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications*, 1-18.

[4] Inbarani, H. H., Azar, A. T., & Jothi, G. (2014). Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. *Computer methods and programs in biomedicine*, *113*(1), 175-185.

[5] Mlakar, U., Fister, I., Brest, J., & Potočnik, B. (2017). Multi-objective differential evolution for feature selection in facial expression recognition systems. *Expert Systems with Applications*, *89*, 129-137.

[6] Sweetlin, J. D., Nehemiah, H. K., & Kannan, A. (2017). Feature selection using ant colony optimization with tandem-run recruitment to diagnose bronchitis from CT scan images. *Computer methods and programs in biomedicine*, *145*, 115-125.

[7] Jain, I., Jain, V. K., & Jain, R. (2018). Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification. *Applied Soft Computing*, *62*, 203-215.

[8] Adeli, A., & Broumandnia, A. (2018). Image steganalysis using improved particle swarm optimization based feature selection. *Applied Intelligence*, 48(6), 1609-1622.

[9] Lin, K. C., Zhang, K. Y., Huang, Y. H., Hung, J. C., & Yen, N. (2016). Feature selection based on an improved cat swarm optimization algorithm for big data classification. *The Journal of Supercomputing*, *72*(8), 3210-3221.

[10] Chen, L. F., Su, C. T., Chen, K. H., & Wang, P. C. (2012). Particle swarm optimization for feature selection with application in obstructive sleep apnea diagnosis. *Neural Computing and Applications*, *21*(8), 2087-2096.

[11] Shang, L., Zhou, Z., & Liu, X. (2016). Particle swarm optimization-based feature selection in sentiment classification. *Soft Computing*, *20*(10), 3821-3834.

[12] ElAlami, M. E. (2011). A novel image retrieval model based on the most relevant features. *Knowledge-Based Systems*, *24*(1), 23-32.

[13] Zawbaa, H. M., Emary, E., Parv, B., & Sharawi, M. (2016, July). Feature selection approach based on moth-flame optimization algorithm. In *Evolutionary Computation (CEC), 2016 IEEE Congress on* (pp. 4612-4617). IEEE.

[14] Ghaemi, M., & Feizi-Derakhshi, M. R. (2016). Feature selection using forest optimization algorithm. *Pattern Recognition*, *60*, 121-129.

[15] Yang, X. S., Deb, S., & Fong, S. (2014). Metaheuristic algorithms: optimal balance of intensification and diversification. *Applied Mathematics & Information Sciences*, *8*(3), 977.

[16] Mafarja, M. M., & Mirjalili, S. (2017). Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing*, *260*, 302-312.

[17] Chuang, L. Y., Chang, H. W., Tu, C. J., & Yang, C. H. (2008). Improved binary PSO for feature selection using gene expression data. *Computational Biology and Chemistry*, *32*(1), 29-38.

[18] Chen, Y. P., Li, Y., Wang, G., Zheng, Y. F., Xu, Q., Fan, J. H., & Cui, X. T. (2017). A novel bacterial foraging optimization algorithm for feature selection. *Expert Systems with Applications*, *83*, 1-17.

[19] Babatunde, O. H., Armstrong, L., Leng, J., & Diepeveen, D. (2014). A genetic algorithm-based feature selection. *International Journal of Electronics Communication and Computer Engineering*, 5(4), 899-905.

[20] Ghamisi, P., & Benediktsson, J. A. (2015). Feature selection based on hybridization of genetic algorithm and particle swarm optimization. *IEEE Geoscience and Remote Sensing Letters*, *12*(2), 309-313.

[21] Wang, G., Chu, H. E., Zhang, Y., Chen, H., Hu, W., Li, Y., & Peng, X. (2015). Multiple parameter control for ant colony optimization applied to feature selection problem. *Neural Computing and Applications*, *26*(7), 1693-1708.

[22] Ahmad, F., Isa, N. A. M., Hussain, Z., Osman, M. K., & Sulaiman, S. N. (2015). A GA-based feature selection and parameter optimization of an ANN in diagnosing breast cancer. *Pattern Analysis and Applications*, *18*(4), 861-870.

[23] Chen, Y., Miao, D., & Wang, R. (2010). A rough set approach to feature selection based on ant colony optimization. *Pattern Recognition Letters*, *31*(3), 226-233.

[24] Eardley DM (1974) Death of white holes in the early Universe. *Phys Rev Lett* 33:442.

[25] Davies PC (1978) Thermodynamics of black holes. Rep Prog Phys 41:1313.

[26] Morris MS, Thorne KS (1988) Wormholes in spacetime and their use for interstellar travel: a tool for teaching general relativity. *Am J Phys* 56:395-412.

[27] Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, *27*(2), 495-513.

[28] Guth AH (2007) Eternal inflation and its implications. *J Phys A Math Theor* 40:6811.

[29] Steinhardt PJ, Turok N (2005) The cyclic model simplified. *New Astron Rev* 49:43–57.

[30] Chuang, L. Y., Yang, C. H., & Li, J. C. (2011). Chaotic maps based on binary particle swarm optimization for feature selection. *Applied Soft Computing*, *11*(1), 239-248.

[31] Banati, H., & Bajaj, M. (2011). Fire fly based feature selection approach. *IJCSI International Journal of Computer Science Issues*, *8*(4), 473-480.

[32] Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X., & Wang, S. (2011). An improved particle swarm optimization for feature selection. *Journal of Bionic Engineering*, *8*(2), 191-200.

[33] Nakamura, R. Y., Pereira, L. A., Costa, K. A., Rodrigues, D., Papa, J. P., & Yang, X. S. (2012, August). BBA: a binary bat algorithm for feature selection. In 2012 25th SIBGRAPI conference on graphics, Patterns and Images (pp. 291-297). IEEE.

[34] Han, X., Chang, X., Quan, L., Xiong, X., Li, J., Zhang, Z., & Liu, Y. (2014). Feature subset selection by gravitational search algorithm optimization. *Information Sciences*, *281*, 128-146.

[35] Emary, E., Zawbaa, H. M., Ghany, K. K. A., Hassanien, A. E., & Parv, B. (2015, September). Firefly optimization algorithm for feature selection. In *Proceedings of the 7th Balkan Conference on Informatics Conference* (p. 26). ACM.

[36] Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, *172*, 371-381.

[37] Wang, H., & Niu, B. (2017). A novel bacterial algorithm with randomness control for feature selection in classification. *Neurocomputing*, *228*, 176-186.

[38] Mafarja, M., Eleyan, D., Abdullah, S., & Mirjalili, S. (2017, July). S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. In *Proceedings of the International Conference on Future Networks and Distributed Systems* (p. 14). ACM.

[39] Ewees, A. A., El Aziz, M. A., & Hassanien, A. E. (2019). Chaotic multi-verse optimizer-based feature selection. *Neural Computing and Applications*, 31(4), 991-1006.

[40] Shunmugapriya, P., & Kanmani, S. (2017). A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid). *Swarm and Evolutionary Computation*, *36*, 27-36.

[41] Aladeemy, M., Tutun, S., & Khasawneh, M. T. (2017). A new hybrid approach for feature selection and support vector machine model selection based on self-adaptive cohort intelligence. *Expert Systems with Applications*, *88*, 118-131.

[42] Nagpal, S., Arora, S., & Dey, S. (2017). Feature Selection using Gravitational Search Algorithm for Biomedical Data. *Procedia Computer Science*, *115*, 258-265.

[43] Faris, H., Hassonah, M. A., Ala'M, A. Z., Mirjalili, S., & Aljarah, I. (2018). A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. *Neural Computing and Applications*, 30(8), 2355-2369.

[44] Aljarah, I., Ala'M, A. Z., Faris, H., Hassonah, M. A., Mirjalili, S., & Saadeh, H. (2018). Simultaneous Feature Selection and Support Vector Machine Optimization Using the Grasshopper Optimization Algorithm. *Cognitive Computation*, 10(3), 478-495.

[45] Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, *62*, 441-453.

[46] Suganthi, M., & Karunakaran, V. (2018). Instance selection and feature

extraction using cuttlefish optimization algorithm and principal component analysis using decision tree. *Cluster Computing*, 1-13.

[47] Dong, H., Li, T., Ding, R., & Sun, J. (2018). A Novel Hybrid Genetic Algorithm with Granular Information for Feature Selection and Optimization. *Applied Soft Computing*, 65, 33-46.

[48] Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation,* 9, 1-14.

[49] Saremi, S., Mirjalili, S., & Lewis, A. (2015). How important is a transfer function in discrete heuristic algorithms. Neural Computing and Applications, 26(3), 625-640.

[50] Xue, B., Zhang, M., & Browne, W. N. (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, *43*(6), 1656-1671.

[51] Hall, M. A. (1999). Correlation-based feature selection for machine learning.

[52] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175-185.

[53] Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary ant lion approaches for feature selection. *Neurocomputing*, *213*, 54-65.

[54] Asuncion, A., & Newman, D. (2007). UCI machine learning repository.

[55] Jensen, R., & Shen, Q. (2008). *Computational intelligence and feature selection: rough and fuzzy approaches* (Vol. 8). John Wiley & Sons.

[56] Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3-18

[57] Reddy, K. S., Panwar, L. K., Panigrahi, B. K., & Kumar, R. (2018). A New Binary Variant of Sine–Cosine Algorithm: Development and Application to Solve Profit-Based Unit Commitment Problem. *Arabian Journal for Science and Engineering*, 43(8), 4041-4056.

[58] Faris, H., Hassonah, M. A., Ala'M, A. Z., Mirjalili, S., & Aljarah, I. (2018). A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture. *Neural Computing and Applications*, 30(8), 2355-2369.

[59] Sohrabi, M. K., & Tajik, A. (2017). Multi-objective feature selection for warfarin dose prediction. *Computational biology and chemistry*, *69*, 126-133.

[60] Zouache, D., & Abdelaziz, F. B. (2018). A cooperative swarm intelligence algorithm based on quantum-inspired and rough sets for feature selection. *Computers & Industrial Engineering*, *115*, 26-36.

[61] Tamimi, E., Ebadi, H., & Kiani, A. (2017). Evaluation of different metaheuristic optimization algorithms in feature selection and parameter determination in SVM classification. *Arabian Journal of Geosciences*, *10*(22), 478.

[62] Tiwari, S., Singh, B., & Kaur, M. (2017). An approach for feature selection using local searching and global optimization techniques. *Neural Computing and Applications*, *28*(10), 2915-2930.

[63] Ekbal, A., & Saha, S. (2016). Simultaneous feature and parameter selection using multiobjective optimization: application to named entity recognition. *International Journal of Machine Learning and Cybernetics*, *7*(4), 597-611.

[64] Lin, S. W., Chen, S. C., Wu, W. J., & Chen, C. H. (2009). Parameter determination and feature selection for back-propagation network by particle swarm optimization. *Knowledge and Information Systems*, *21*(2), 249-266.

[65] Vijaya, J., & Sivasankar, E. (2017). An efficient system for customer churn prediction through particle swarm optimization based feature selection model with simulated annealing. *Cluster Computing*, 1-12.

[66] Manikandan, R. P. S., & Kalpana, A. M. (2017). Feature selection using fish swarm optimization in big data. *Cluster Computing*, 1-13.

[67] Emary, E., & Zawbaa, H. M. (2019). Feature selection via Lèvy Antlion optimization. *Pattern Analysis and Applications*, 22(3), 857-876.

[68] Tegmark, M. (2004). Barrow, JD Davies, PC Harper, CL, Jr eds. *Science and Ultimate Reality Cambridge University Press Cambridge*.

[69] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2010). BGSA: binary gravitational search algorithm. Natural Computing, 9(3), 727-745.

[70] Bagga, P., Hans, R., & Sharma, V. (2017). N-grams Based Supervised Machine Learning Model for Mobile Agent Platform Protection against Unknown Malicious Mobile Agents. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(6), 33-39.

[71] H. M. Keerthi Kumar, B. S. Harish. Automatic Irony Detection using Feature Fusion and Ensemble Classifier, *International Journal of Interactive Multimedia and Artificial Intelligence*, (2019).

[72] Revanasiddappa, M. B., & Harish, B. S. (2018). A new feature selection method based on intuitionistic fuzzy entropy to categorize text documents. *International Journal of Interactive Multimedia and Artificial Intelligence, 5*(3), 106-117.

[73] Pujari, D., Yakkundimath, R., & Byadgi, A. S. (2016). SVM and ANN based classification of plant diseases using feature reduction technique. *International Journal of Interactive Multimedia and Artificial Intelligence*, 3(7), 6-14.

[74] Belkhodja, L., & Hamdadou, D. (2019). IMCAD: Computer Aided System for Breast Masses Detection based on Immune Recognition. *International Journal of Interactive Multimedia and Artificial Intelligence*, 5(5), 97-108.

Rahul Hans

Rahul Hans has received his B.Tech degree in Computer Science and Engineering from Punjab Technical University, Jalandhar and M.Tech. degree in Computer Science and Engineering from Guru Nanak Dev University, Amritsar. He is currently pursuing Ph.D. from Guru Nanak Dev University, Amritsar. His predominant research areas include Mobile Agent Systems, Machine Learning and Metaheuristics. He has published research papers in various reputed international conferences and journals.

Harjot Kaur

Harjot Kaur is presently working as Assistant Professor in the Department of Computer Science and Engineering at Guru Nanak Dev University, Regional Campus Gurdaspur, Punjab, India. She has completed her Ph.D. degree from the Department of Computer Science and Engineering, Guru Nanak Dev University, Amritsar, Punjab, India. Her predominant area of research is Distributed Artificial Intelligence. She has published research papers in various reputed International Journals. Also she has attended various national and international Conferences and presented many research papers.

Comparative Study on Ant Colony Optimization (ACO) and K-Means Clustering Approaches for Jobs Scheduling and Energy Optimization Model in Internet of Things (IoT)

Sumit Kumar[1], Vijender Kumar Solanki[2]*, Saket Kumar Choudhary[3], Ali Selamat[4,5], Rubén González Crespo[6]

[1] Department of Information Technology, Gopal Narayan Singh University, Sasaram, Bihar (India)
[2] Department of Computer Science & Engineering, CMR Institute of Technology, Hyderabad, TS (India)
[3] Saket Kumar Choudhary, FCA, MRIIS, Faridabad, Haryana (India)
[4] Media and Games Center of Excellence (MagicX) Universiti Teknologi Malaysia & School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Skudai, Johor (Malaysia)
[5] Malaysia Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia Kuala Lumpur, Jalan Sultan Yahya Petra, Kuala Lumpur (Malaysia)
[6] Department of Computer Science and Technology, Universidad Internacional de La Rioja, Logroño (Spain)

Received 4 November 2019 | Accepted 22 January 2020 | Published 27 January 2020

Abstract

The concept of Internet of Things (IoT) was proposed by Professor Kevin Ashton of the Massachusetts Institute of Technology (MIT) in 1999. IoT is an environment that people understand in many different ways depending on their requirement, point of view and purpose. When transmitting data in IoT environment, distribution of network traffic fluctuates frequently. If links of the network or nodes fail randomly, then automatically new nodes get added frequently. Heavy network traffic affects the response time of all system and it consumes more energy continuously. Minimization the network traffic/ by finding the shortest path from source to destination minimizes the response time of all system and also reduces the energy consumption cost. The ant colony optimization (ACO) and K-Means clustering algorithms characteristics conform to the auto-activator and optimistic response mechanism of the shortest route searching from source to destination. In this article, ACO and K-Means clustering algorithms are studied to search the shortest route path from source to destination by optimizing the Quality of Service (QoS) constraints. Resources are assumed in the active and varied IoT network atmosphere for these two algorithms. This work includes the study and comparison between ant colony optimization (ACO) and K-Means algorithms to plan a response time aware scheduling model for IoT. It is proposed to divide the IoT environment into various areas and a various number of clusters depending on the types of networks. It is noticed that this model is more efficient for the suggested routing algorithm in terms of response time, point-to-point delay, throughput and overhead of control bits.

Keywords

Ant Colony Optimization (ACO), Energy Consumption, Internet of Things (IoT), K-means Algorithm, Response Time, Message Scheduling.

DOI: 10.9781/ijimai.2020.01.003

I. Introduction

THE latest IoT applications depend on promotion of wireless sensor networks (WSNs) with expert of engineering. These IoT applications contain a large number of devices, connected with different requirements and technologies. Such kinds of IoT applications do the sensing and collection of data with transmission of data to the administrator nodes for other possible operations and even a cloud at the backdrop for data analytics. These processes require routing protocols for their completion. Routing protocols have two major challenges. The first challenge is to improve data transmission and scalability whereas the second challenge is to minimize energy consumption. In an IoT application, network nodes under different network topology collect different kind of data so that an IoT application produces an enormous amount of data. The heterogeneity in network topology restricts the TCP/IP to become the best policy for proper resource allocation to computing and routing [1]-[3], [27]-[29].

Owing to the above-mentioned challenges, different persons view IoT in different ways, based on their perception and requirements. A routing protocol includes the multiple job scheduling methodologies. These job scheduling methodologies are reported as either heuristic or metaheuristic-based approaches. Heuristic-based methodologies are comparatively more helpful when we look for a local optimum whereas metaheuristic methodologies further try to explore the solution

* Corresponding author.
E-mail address: spesinfo@yahoo.com

space to attain global optima. Despite the fact that the metaheuristic methodologies look very engaging and a large number of parameters to be turned on account of IoT thus limits the utilization [4]-[9], [27]-[29].

A number of researchers have developed and used the ACO algorithms for finding the shortest path in several routing problems. An ACO algorithm includes a stochastic local search strategy to structure the routing paths which can be established by a set of artificial ants. These ants work cooperatively using indirect communication of information for construction of the optimal shortest path. Inan intelligent optimization algorithm, the ACO idea is borrowed from the food searching characteristic of the real ant colony and how ants do this difficult job when they work together. Depending on biological studies on ants, it can be assumed that the ACO performs the finding of the shortest path from the nest to the food. Ant's pheromone distribution mechanism to share information with other ants in indirect coordination is called stigmergy. A number of researchers have suggested that the ACO optimization algorithm is very good for collaborating, exchange and transmission of information. The ACO algorithm is based on pheromone updates. This pheromone updating depends on the best solution achieved by the pheromone amount and the number of ants. The natural ants find the shortest path based on their own best knowledge solution and it depends on a strong pheromone trace. Finding the shortest path using ACO algorithm is inversely proportional to the pheromone quantity and length of the path [10]-[12].

The ACO depends on a probabilistic method for solving the computational problems and minimizes the paths through graphs [13]. ACO algorithm can be given in detail as:

Let us assume

K = Number of ants

(Tau) ι_{ij} = The pheromone concentration with the edge

η_{ij} = The heuristic information (based on experience)

α = pheromone removal parameter

β = pheromone deposition parameter

ρ = Represents the evaporation rate to avoid accumulation of the pheromones

$\Delta\tau_{i,j}^{k}$ = Amount of pheromones needed to path travel by ant K

L_k = Total length travel by ants K

Q = Constant

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\iota_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{k\in allowed k}[\iota_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} & \text{If j } \varepsilon \text{ allowed k} \\ 0 & \text{Otherwise} \end{cases}$$

$$\iota_{ij}(t+n) = \rho\iota_{ij}(t) + \Delta\iota_{ij}$$

$$\Delta\iota_{i,j}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{If (i, j) } \varepsilon \text{ tour described by taboo k} \\ 0 & \text{Otherwise} \end{cases}$$

Clustering is used in a wide range of research areas like engineering, medicine, data mining, biology, artificial intelligence and even IoT. Xu and Wunsch (2005) have represented an abbreviated survey on clustering algorithms. K-means clustering is the most commonly used algorithm. K-means clustering algorithm divides the data/substances into a number of clusters based on minimizing the sum of the squared distances between the data/substances and the centroid of the clusters. The k-means clustering algorithm is one of the simplest algorithms, but it is not suitable for a large amount of data set due to higher time complexity. Various methods have been proposed to accelerate the working of k-means such as when computation complexity is increased the backtracking is required [4], [12].

In recent years, many new clustering algorithms have been proposed after deep study on clustering. There is a clustering algorithm which is based on ant system. A combination of two or three different clustering algorithms is new to clustering analysis. Clustering analysis plays an important role in the datamining field. Data can be grouped into different classes or clusters by clustering analysis. There exists better similarity among the objects in the same class and poorer similarity among the objects in different classes. In machine learning, clustering is a kind of unsupervised learning because it has no prior knowledge of classification labels. Clustering analysis is widely applied in image processing, model recognition, document retrieval, medical diagnosis, web analysis etc. [4], [14].

A. The Basic Principle of the K-means Algorithm

K objects are randomly chosen from n objects as initial clustering centers. Then the algorithm calculates the distance from each object to k clustering centers and judges which clustering center is nearest, assigning the object to the cluster of the nearest center. When all the computation work is done, it will form knew clusters. Next, the algorithm re-computes a mean value of each new cluster as its new clustering center. According to the above procedure, the algorithm will repeat calculating the distance and iterating till criterion function converges. The sum of square error is often-used as the criterion function. It is defined as [4], [14]:

$$E = \sum_{i=1}^{k}\sum_{p=Ci}|p - mi|2 \quad (1)$$

E is the sum of square error to all objects in the database. p is a point in the space that expresses a given object. m_i is the mean of clustering C_i. According to this criterion, data belonging to the same class are as similar as possible and data from different classes are as different as possible [4], [14].

The article is divided into six sections. After a brief introduction about ACO and k-means clustering in section I, section II contains the related work. In section III we explain our problem definition, in section IV we show our proposed algorithm, and finally, section VI explains the discussion and conclusions.

II. Related Work

The IoT environment contains a large scale of different types of networks. Routing techniques in WSN from source to destination is one of the important issues in the IoT system. The algorithms which are used to select the cluster heads/nodes depend on specific characteristics of clusters and/or network environments like energy level that suffers from complexity. Hence, the architectures of IoT are unsuitable and features of IoT application are dynamic. A large number of different kinds of research work on routing in IoT from source to destination have been done in literature [1], [4], [26].

Omar Sajid [1] has proposed optimum routing path using Ant Colony Optimization (ACO) algorithms inside the IoT system. Depending on the types of network, Sajid [1] has suggested to divide the IoT environment into various zones like status, requirements, etc. then use the ACO algorithm that was fit for each network. Finally, the simulation results proved that the proposed routing algorithm has better energy saving techniques. Kumar et al. [4] has presented a comparison of some clustering algorithms to analyze the scheduler performance and suggested that K-means based clustering is effective for the IoT based environment. Lu et al.[10] has suggested that the ACO finds the path to broadcast signaling contained in various network nodes and various flexible network structure problems, and during simulation analysis it is noticed that finding the path by ACO in IoT decreases the transmission storm efficiently. When the number of nodes increases in the finding path process, then it is important to reduce the time of path structure. In order to analyze a large-scale routing strategy, Guang Ji [15] has

proposed IoT ant colony searching routing based on Markov decision model. Markov decision ant colony routing selection algorithm is based on multi-parameter equilibrium. Markov routing is a decision model to estimate the number of nodes in a node communication range and facilitates the decision which meets the requirements. This algorithm efficiently decreases the overhead workload which is generated by controlled messages and multiple hops routing between clusters and make the evaluation function value of the path for allowing decision set corresponding to the pheromone concentration of ant colony for repeating the process. During the routing discovery phase, it calculates the transition probability of nodes and selects the global optimal routing. During the simulation-based analysis, it is observed that the problem of network "hot spots" is effectively solved by the Markov-A algorithm and the energy consumption of the network is balanced so that the life cycle of the network is prolonged. Dorigo et. al. [16] has specified an explanation of the Ant Colony Optimization (ACO) meta-heuristic and has discussed the type of problems where it can be applied. Dorigo et. al. [16] has used ACO algorithms in two typical applications, namely traveling salesman problem and routing in packet-switched networks. Merkle et. al. [17] has introduced an ACO method for the resource-constrained project scheduling problem (RCPSP). It is a combination of direct (or local) and indirect (or global) for ants in the structure of a new solution and uses pheromone evaluation approaches. From newly added features, this algorithm changes the strength of the heuristic effect and the rate of pheromone vaporization over ant peers. Below some limitation author's proposed algorithm perform the best solution compared to some other heuristics with and without limitations to the number of evaluated schedules shows the flexibility of the method. Michael Frey et al. [18] have proposed a framework and methodology to study ant routing algorithms for wireless networks. While running experiments in a wireless test bed is a number of some, expensive and error-free task, studying ant routing algorithms in simulation allows investigating some specifics properties of these algorithms more easily. This includes behavior of all aspects such as adaptive and pheromone evolution, the scalability in respect to the number of nodes or traffic flows, and mobile scenarios. These frameworks are easy to extend and customize by providing new back ends for different network simulators (or test bed frameworks) which is feasible with acceptable efforts. Mariusz et al. [19] have proposed Ant Colony Optimization (ACO) based algorithm designed to find the shortest path in a graph. The algorithm consists of several sub problems that are presented successively. Each sub problem is discussed from many points of view to enable researchers to find the most suitable solutions for the problems. Algorithms based on the metaheuristic of ant colony do not guarantee finding an optimal solution in all possible cases. Accordingly, to experimentation, it is particularly important to find out and select parameters dedicated to each of the problems under consideration. Individual elements of the procedures applied in the process should be also analyzed with regarding to their usability and purpose fullness of application. The construction of this Shortest Path ACO algorithm directly reflects to the various variants of the execution of individual elements of the procedure. In this way, it is possible to improve the method for the solution of the shortest path problem to approach or reach optimal solutions. An evaluation of the duration time and the quality of returned solutions will provide information for making a decision on the implementation of a given scheme as being of optimum quality or an alternative to more time-consuming procedures or procedures with higher computational cost. Yuq in get. al. [20] has recommended a new K-Means algorithm. This algorithm is a combination of density-based and ant searching theory, which is controlled by the initial parameter of k-means and local minimal by the random ants. The experiments analysis shows that, this algorithm has better quality for productivity and accuracy of clusters. Thus, it can place the similar types of objects together in one cluster and eliminate the dissimilar types of objects away. This procedure has random competence of ACO which avoid clustering success into local optimality, and it furthermore avoids responsiveness of the primary partition of the k-means algorithm. Gelenbe et al. [25] has proposed the relationship and effect between choice of system load, energy consumption and QoS using a simple queuing model. They [25] have analyzed the parameters which are effect of response time of the system and energy cost per job using the mathematical queuing model. S. Kumar et Al. [30], [31] and V. García-Díaz et Al. [32] have proposed the Supply Chain Management based model for optimizing the response time and job scheduling by applying M/M/1 queuing model in IoT environment.

III. The Problem Definition

This section contains the problem statement followed by a description of the ACO and K-Means clustering based IoT messaging service architecture considered in the work and the job model.

A. Problem Statement

In an IoT environment, a big amount of heterogeneous wired and wireless devices/objects interconnect with each other identified by IPv6 addressing using single or multiple levels of subnets. These devices/objects generate a big amount of data (much time a continuous stream too) and scheduling these in the IoT environment from source to destination becomes a challenging issue.

IoT is a mixture of multiple wired or wireless communication technologies. The routing is the most important challenge in the IoT environment for solving how to find the best optimal path for data transmission from one node to another node in a different environment. An IoT environment includes different types of networks which depend on the network's status, and requirements. In this article, it is proposed that each network has own responsibility for finding an optimal path in the IoT environment. There are many inter-connections between different networks and the seinter-connections are called overlapped areas. So, this work intends creating an algorithm to control the use of algorithms and determines a solution for overlapped areas problem. This algorithm has been tested and compared with ACO and K-Means clustering algorithms that are closely related to the IoT routing problem.

The prime objective of this work is to calculate and compare the response time for message forwarding of the entire IoT environment using ACO and k-means clustering algorithms approaches to find the suitable path for reducing the energy consumption/cost. In the big transportation for IoT background, this is very beneficial for providing the flexible and effective response time services. The number of clusters is one of the most important features for calculating the performance of K-means clustering algorithm and the number of paths is one of the most important features for calculating the performance of ACO algorithm. The nodes (for ACO algorithm) and Cluster Head (*CH*) (for K-means clustering algorithm) may be completely connected or partially connected. The processing speed of each *CH/Node* can be measured as MIPS (Million Instruction Per Cycle) count [4], [21], [26].

B. Job Model

Each job or message is divided into sub-jobs/tasks depending on the priority of jobs and sequence of data messages. Data are available in the format of data packets and need the transmission from source to destination. The jobs are mathematically modeled by a weighted graph $D= (T, E)$ where T shows the set of t tasks and E shows as the set of e edges among the jobs. The edges show the priority of task/message [4], [26].

C. A Route Planning Model for K-Means Clustering and ACO Algorithms

The performance measurement for both K-Means clustering algorithms and ACO algorithms is done in terms of the average response time. It is assumed that each server follows the *M/M/1* queuing model where λ is the arrival rate and μ is the service rate [4], [22], [26].

The set of objects in different colors represents them belonging to different clusters and paths. A sample shortest path is shown between source to destination base station based on a hypothetical approach.

QoS parameters like Average Response Time (*RT*), Average Waiting Time (*WT*), and Average Queue Length (*QL*) have been estimated. Here, it is assumed that each server follows the *M/M/*1 queuing model. The average queue length at the i_{th} *CH/Node* with both the number of jobs waiting in the queue and those in service can be written as $E\,[Ni] = \frac{\rho_i}{(1-\rho_i)}$ Where ρ_i is the i_{th} server utilization and whose average queue length is *E [Ni]* [4], [23]-[24], [26].

The average response time depends on how to quick response a *CH/Node*. Arrival rate of the jobs at the input queue at each *CH/Node* is randomly generated. The average response time can be estimated as [4], [23]-[24], [26]:

$$\text{Average Response Time} = \frac{\text{AveregeQueueLength}}{\text{ArrivalRate}} \quad (2)$$

The waiting time is the period of time where the job does not execute because of the execution preference or some event to happen. So, the average waiting time at each CH/Node in the path can be calculated as [4], [23]-[24], [26]:

$$\text{Average Waiting Time} = \text{AverageResponseTime} - \frac{1}{\text{ServiceRate}} \quad (3)$$

Fig. 1, represents a short example of the proposed IoT routing algorithm model using ACO algorithm.

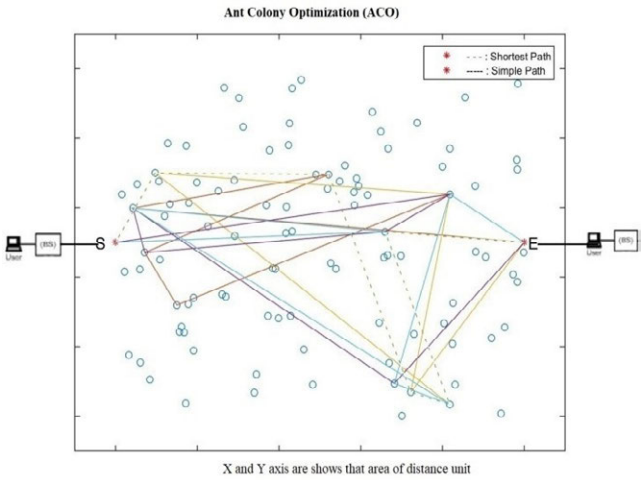


Fig. 1. Architecture of IoT Network Using ACO Algorithm and Messaging Forwarding.

Let us assume that there are *n* nodes in a network and *m* nodes transmitting the signal for searching at the same time in network routing. $\iota_{ij}(t)$ is the number of active signaling established through the path among node *i* and *j*, d_{ij} *(i, j = 1, 2,...,n)*stands for the distance among the node *i* and *j* at the time *t*.

In the initiation stage, m random nodes are selected, the number of active signaling among the nodes *i* and *j* are $\iota_{ij}(0)$ and ι *(tabu)* is the primary part of individual signaling. *k* is allocated as a preliminary node.

Here, ι_{ij}^{k} *(t)* stands for the probability of *k* transmitting signal from node *i* to node *j* at the time of *t*, then:

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\iota_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{k\in \text{allowed } k}[\iota_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} \text{ If j } \varepsilon \text{ allowed } k \\ 0 \qquad \text{Otherwise} \end{cases} \quad (4)$$

The permitted *k= {0, 1... ,n - 1}*, and ι^{st} and for the set of nodes and indicating next permitted node to pass. The difference between an artificial ant and real ant colony is the capability of memory. The ι (*k = 1, 2,...,m*) is utilizing the records of nodes and signaling *k* pass over at the current time, and it is dynamically adjusted with the transmission of signaling *k* process. *1-ρ*represents the unit of disappearing and *α, β*individually represent the amount of information collection of signaling in the process of re-transmission. It plays different roles for the heuristic aspect in the path selection through the signaling re-transmission. $\eta_{ij}(t)$ represents the predictions unit of the transmission between node *i* to *j*. Then signaling *k* cover all nodes and make a complete cycle. At that time, the information of all paths is updated according to the following equation:

$$\iota_{ij}(t+n) = \rho\, \iota_{ij}(t) + \Delta\, \iota_{ij} \quad (5)$$

$$Then\ \Delta\, \iota_{ij} = \sum_{k=1}^{m} \Delta\iota_{ij}^{k} \quad (6)$$

$\Delta\iota_{ij}^{k}$ represents the amount of information of signaling *k's* suggestions between node *i* and *j* in this cycle.

$$\Delta\iota_{i,j}^{k} = \begin{cases} \frac{Q}{L_k} \text{ if signaling k passes through the node i and j in the cycle} \\ 0 \qquad \text{Otherwise} \end{cases} \quad (7)$$

Here, *Q* represents the constant, L_k represents the path length that signaling k has paced in this cycle.

In order to estimate the shortest path using ACO from a source node (*S*) to a destination (*E*) it is assume that λ_1 isthe arrival rate of the job at $Node_1$, working as a source node (*S*) for transmission to the destination (*E*)$Node_5$. $Node_1$ has nodes $Node_2$, $Node_3$, and $Node_4$ as immediate neighbors. Let the probability of arrival of jobs at queues of these neighbors be P_2, P_3 and P_4 respectively such that $P_2+P_3+P_4=1$. In this article it is assumed that there are equal probabilities for selection of all the paths i.e. $P_2=P_3=P_4$. After being serviced by server $Node_1$, the jobs arrive at $Node_2$ being the best path among the available paths offering the minimum response time with probability P_3. Let this probability be referred to as $P_1$for the remaining path, indicating the path chosen in the beginning. Therefore, the arrival rate for $Node_3$ can be written as $\lambda_1 P_3 = \lambda_1 P_1$. Similarly, after being serviced by server $Node_3$, the jobs arrive at $Node_4$ followed by $Node_3$ to finally reach $Node_5$ with arrival rate $\lambda_1 P_1$.

Let the service rate of $Node_1$, $Node_2$, $Node_3$, $Node_4$, and $Node_5$beμ_1, μ_2, μ_3, μ_4*and* μ_5 respectively.

Therefore, the utilization of $Node_1$ with $P_1=1$ becomes

$$\rho_1\,(Node_1) = \frac{\lambda_1}{\mu_1} \quad (8)$$

The utilization of the selected nodes can be written as [4], [23]-[24], [26]

$$\rho_j\,(Node_j) = \frac{\lambda_1 P}{\mu_j} \quad (9)$$

Where *i* = Number of stages in the network and

j = Index of the selected node in each stage.

The average queue length for $Node_1$ becomes

$$E\,[QL_1^{(Node_1)}] = \frac{\rho_1(Node1)}{(1-\rho_1\,(Node1))} \text{(Using } \rho_1\,(Node_1) \text{ value from Eq. (8))}$$

$$= \frac{\frac{\lambda_1}{\mu_1}}{1-\frac{\lambda_1}{\mu_1}} = \frac{\frac{\lambda_1}{\mu_1}}{\frac{\mu_1-\lambda_1}{\mu}} = \frac{\lambda_1}{\mu_1-\lambda_1} \quad (10)$$

Similarly, the average queue length can be calculated for the other selected nodes in the path as [4], [23]-[24], [26]

$$E\,[QL_j^{(Node_j)}] = \frac{\rho_{1j}(Nodej)}{(1-\rho_j\,(Nodej))} = \frac{\lambda_1 P}{\mu_j - \lambda_j} \quad (11)$$

The queue length for the entire path can be estimated as:

$E\,[QL_{avg}]$ = Sum of Average Queue Length (QL) value of selected nodes in every stage.

Therefore,

$$E\,[QL_{avg}] = \sum E\left[QL_i^{Nodej}\right] \quad (12)$$

Here 'i' is the stage and $Node_j$ is the node being selected for message forwarding.

$$Average\ Response\ Time = \frac{AveregeQueueLength}{ArrivalRate} \text{(from equation (2))}$$

In the current case, the path from $Node_1$ (source) to $Node_5$ (destination) comprises of $Node_2$, $Node_3$, and $Node_4$. The average response time of for $Node_1$:

$$E\,[RT_1^{(Node_1)}] = \frac{E\,[N_1(Node1)\,]}{\lambda_1} \text{(using } E\,[QL_1\,(Node_1)] \text{ value from Eq. (10))}$$

$$= \frac{\frac{\lambda_1}{\mu_1-\lambda_1}}{\lambda_1} = \frac{1}{\mu_1-\lambda_1} \quad (13)$$

For $Node_j$ server the response time can be written as:

$$E[RT_i^{(Nodej)}] = \frac{\rho_i(Nodej)}{(1-\rho_i\,(Nodej))} = \frac{1}{\mu_j-\lambda_1 P} \quad (14)$$

The average response time for the paths selected become the sum of the Average Response Time (RT) values of selected Node $_j$ in every stage 'i'.

$$E[RT_{avg}] = \sum E\left[RT_i^{Nodej}\right] \quad (15)$$

$$Average\ Waiting\ \ Time = AverageResponseTime - \frac{1}{ServiceRate}$$

(from Eq. (3))

$E\,[WT_1^{(Node_1)}] = E\,[R_1^{(Node_1)}] - \frac{1}{\mu_1}$ *(using* $E\,[RT_1\,(Node_1)]$ value from Eq.*(13))*

$$= \frac{1}{\mu_1-\lambda_1} - \frac{1}{\mu_1} = \frac{\mu_1-\mu_1+\lambda_1}{(\mu_1-\lambda_1)\mu_1} = \frac{\lambda_1}{(\mu_1-\lambda_1)\mu_1} \quad (16)$$

Similarly, for Node$_j$ server the waiting time can be written as:

$$E\,[WT_i^{(Nodej)}] = \frac{\rho_i(Nodej)}{(1-\rho_i\,(Nodej))} = \frac{\lambda_1 P}{(\mu_j-\lambda_1 P)\mu_j} \quad (17)$$

The average waiting time for the complete path can be written as the sum of the average Waiting Time (WT) values of selected nodes in every stage as:

$$E[WT_{avg}] = \sum E\left[WT_i^{Nodej}\right] \quad (18)$$

Fig. 2 represents a short example of a proposed IoT routing algorithm model using K-Means clustering algorithms.

In order to use the K-Means clustering algorithms, a sample path chosen among the clusters to route the messages from Source (S) to Destination (E) has been presented and routing paths as per the K-Means clustering algorithmic characteristic and some properties like Euclidean distance and Degree of nodes to find the shortest path from Source (S) to Destination (E).

Here it is assumed that P is the probability of arrival of the jobs at the source queue to be forwarded to the destination in K-Means clustering through Cluster Head (CH). Let the arrival rate of the jobs at CH_1 be λ_1 which is the Start CH 'S' in the IoT network. The packet needs to be transmitted to the end cluster 'E' which is CH_5 which can be the preferred endpoint like cloud storage. After being serviced by CH_1, let the message be forwarded to CH_2. As discussed above, the arrival rate for CH_2 becomes $\lambda_2 P = \lambda_1 P$. Similarly, the message reaches destination CH_5 with arrival rate $\lambda_1 P$, being routed through various cluster heads CH forming the path.

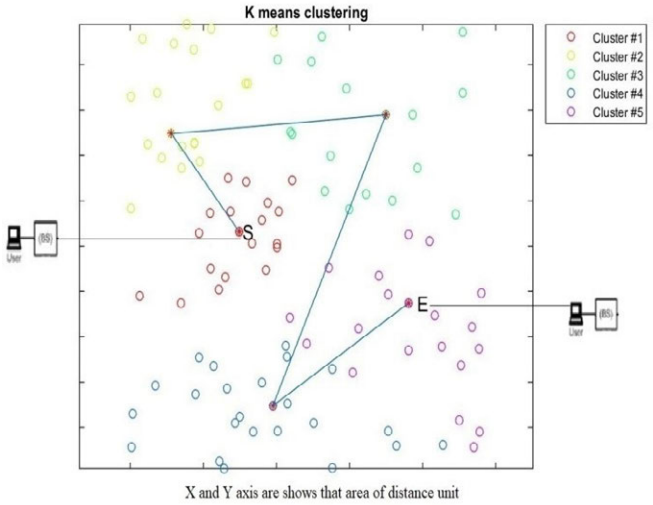


Fig. 2. Architecture of IoT Network Using K-means Clustering Algorithm and Messaging Forwarding.

The best and stable utilization of CH_1 server is [4], [22], [26]

$$\rho_1(CH_1) = \lambda_1/\mu_1 \quad (19)$$

Similarly, for CHj server, the utilization can be written as

$$\rho_j(CH_j) = \lambda_1/\mu_j \quad (20)$$

where i = Number of stages in the network and

j = Index of selected node in each stage.

Therefore, the utilization of $Node_1$ with $P_1 = 1$ becomes $\rho_1\,(Node_1) = \frac{\lambda_1}{\mu_1}$ from Eq.*(8)*.

The average queue length for CH_1 [4], [22], [26]:

$$E\,[QL_1(CH_1)] = \frac{\rho_1(CH1)}{(1-\rho_1\,(CH1))} \text{ (using } \rho_1(CH_1) \text{ value from Eq.(16))}$$

$$= \frac{\frac{\lambda_1}{\mu_1}}{1-\frac{\lambda_1}{\mu_1}} = \frac{\frac{\lambda_1}{\mu_1}}{\frac{\mu_1-\lambda_1}{\mu_1}} = \frac{\lambda_1}{\mu_1-\lambda_1} \quad (21)$$

Similarly, the average queue length can be calculated for the other selected CH_j in the path as [4], [23]-[24], [26]:

$$E\,[QL_j^{(CH_j)}] = \frac{\rho_j(CHj)}{(1-\rho_j\,(CHj))} = \frac{\lambda_1 P}{\mu_j-\lambda_j} \quad (22)$$

The queue length for the entire path can be estimated as:

$E\,[QL_{avg}]$ = Sum of Average Queue Length (QL) value of selected CH in every stage.

Therefore,

$$E\,[QL_{avg}] = \sum E\left[QL_i^{CHj}\right] \quad (23)$$

Where 'i' is the cluster and CH_j is the node being selected for message forwarding.

$$Average\ Response\ Time = \frac{AveregeQueueLength}{ArrivalRate} \text{ (from Eq. (2))}$$

$$E\,[RT_1CH_1)] = \frac{E\,[QL1(CH1)]}{\lambda_1} \text{(using } E[QL_1(CH_1)] \text{ value from Eq. (21))}$$

$$= \frac{\frac{\lambda_1}{\mu_1-\lambda_1}}{\lambda_1} = \frac{1}{\mu_1-\lambda_1} \quad (24)$$

Similarly, average response time can be calculated for the other selected CH_j in the path as [4], [23]-[24]:

$$E[RT_i^{(CHj)}] = \frac{\rho_i(CHj)}{(1-\rho_i\,(CHj))} = \frac{1}{\mu_j-\lambda_1 P} \tag{25}$$

The average response time for the paths selected become the sum of the Average Response Time *(RT)* values of selected CH_j in every stage '*i*'.

$$E[RT_{avg}] = \sum E\left[RT_i^{CHj}\right] \tag{26}$$

$Average\ Waiting\ Time = AverageResponseTime - \frac{1}{ServiceRate}$ (from Eq. (*3*))

$E\,[WT_1^{(CH_1)}] = E\,[R_1^{(CH_1)}] - \frac{1}{\mu_1}$ *(using* $E\,[RT_1\,(CH_1)]$ value from Eq.*(24))*

$$= \frac{1}{\mu_1-\lambda_1} - \frac{1}{\mu_1} = \frac{\mu_1-\mu_1+\lambda_1}{(\mu_1-\lambda_1)\mu_1} \tag{27}$$

Similarly, for CH_j server the waiting time can be written as:

$$E\,[WT_i^{(CHj)}] = \frac{\rho_i(CHj)}{(1-\rho_i\,(CHj))} = \frac{\lambda_1 P}{(\mu_j-\lambda_1 P)\mu_j} \tag{28}$$

The average waiting time for the complete path can be written as the sum of the average Waiting Time (*WT*) values of selected *CH* in every stage as:

$$E[WT_{avg}] = \sum E\left[WT_i^{CHj}\right] \tag{29}$$

IV. The Algorithm

The primary objective of this algorithm is to calculate and compare the message response time from source to destination by preserving the service quality of fluctuating sized messages from the source to the destination through *CHs/Nodes* in the network. The word Cluster Head (*CH*) for K-Means clustering algorithms and the word Node for ACO Algorithms are used. When a sensor device sends a message from the source *CH/Node* to the destination *CH/Nodes*, it fritters away some communication cost in the transfer of the message. In this procedure, it waits for the response of each *CH/Node* in each IoT network environment. This is a big problem for IoT because it may affect the battery life of the sensor *CH/Node* as the battery life gets weak with every waiting period. However, if this response time was reduced by proper path choice, it would decrease the waiting time and eventually preserve the power of the sensor's devices. Thus, this method decreases the response time and makes it an energy-aware scheduling algorithm too.

These messages convey data with respect to the physical parameters which change continuously in some degree. Algorithm 1 corresponds to the message-scheduling algorithm for the IoT framework.

V. Performance Evaluation

To evaluate the performance of QoS parameters, Matlab64-bit version *8. 5.0.197613 (R2013a)*, processor Intel *(R) Core (TM) i7-4790 CPU @ 360GHz, 64*-bit Operating System, and RAM *4 GB* as a simulation platform has been used. *M/M/1* priority queuing model has been used for resource provisioning and message scheduling system within the IoT environment. Applied *M/M/1* queuing model can be reconsidered for some newer models like phase-type queuing networks or Pareto-distribution-based queuing networks. The basis of considering the *M/M/1* queuing model is that it considers each *node/CH* as a single node and has been established to fit well in these types of problems. Random data values have been generated for the experiment during execution.

Algorithm 1: Response Time Aware Scheduler

Initialize:

Submit the message request

Identify the Source 'S' and the Destination 'E'

Record the Arrival rate of the batch of jobs/message

For every Source Node/CH,

Evaluate the number of messages/ modules

Sort the job/messages according to priority

Establish the Queues:

For the Source 'S', do,

Assign job/message to the high priority/low priority queue

Route Selection:

For the source node, select the nearest node using ACO and K-means approach as per Section 3

For the specified job/message

Evaluate the execution time of every job/message at individual CH/ Node.

If a CH/Node is Busy then // Upcoming message is waiting

For i = 2: N // Following messages

Message Wait (i) = Job Execution Time – Priority Message (i, 1);

End for

Calculate Average Queue Length // using eq. 12 (For ACO) and 23 (For K-means)

Calculate Average Response Time// using eq. 15 (For ACO) and 26 (For K-means)

Calculate Average Waiting Time// using eq. 17 (For ACO) and 29 (For K-means)

Fig. 3 represents the Response time of ACO and K-means clustering algorithm for a fixed job size of *1000* and varying the CHs (For K-means) and Node (For ACO) from *10* to *50* with the numeric data represented in Table I. Further, ACO Response time result is a minimum and better than the K-means Clustering Algorithm response time.

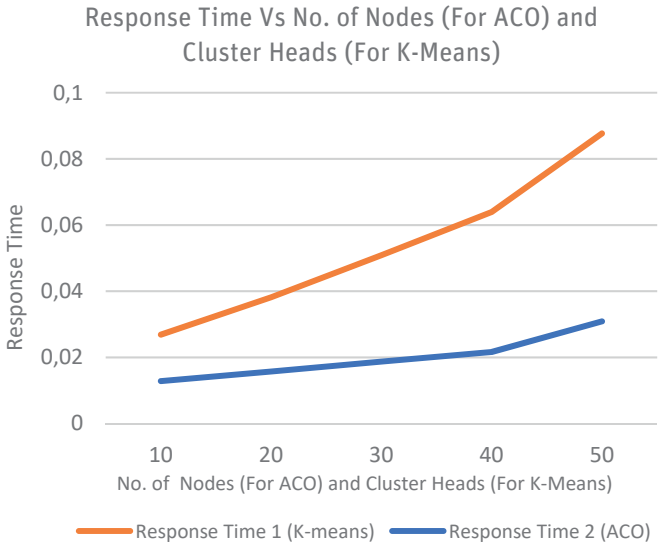


Fig. 3. Batch of Jobs Response Time fora fix jobs size and varying the Nodes and Cluster Heads (CHs).

TABLE I. Response Time for a Fix Jobs Size and Varying the Nodes and Cluster Heads (CHs) Corresponding to Fig. 3

No. of Cluster Heads	Response Time (ACO Algorithm)	Response Time (K-means Clustering Algorithm)
10	0.0128524	0.0140297
20	0.0157312	0.0224375
30	0.018759	0.0321506
40	0.02162207	0.042322676
50	0.03088497	0.056817971

The effect of some other QoS parameters such as the Average Service Rate, Average Queue Length and Average Waiting Time are shown in Fig. 4-6.In this experimental setup, we fixed the job size of *1000* while varying the number of CHs (For K-means) and Node (For ACO)from *10* to *50*. The numeric data is represented in Table II-IV. Further, ACO results are better than the K-means Clustering Algorithm.

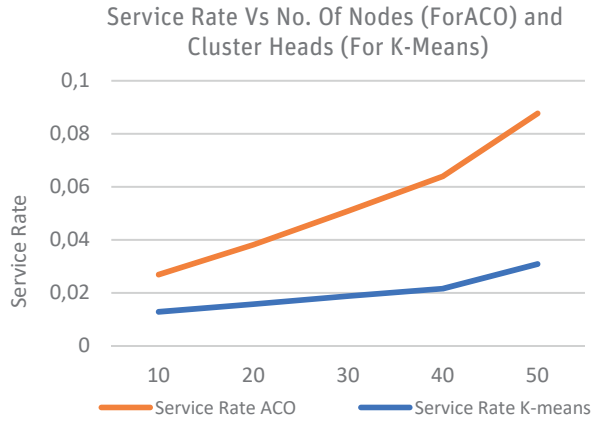


Fig. 4. Batch of Job's Service Rate for a fixed job size and varying Nodes and Cluster Heads (CHs).

TABLE II. Service Rate for A Fixed Job Size and Varying Nodes and Cluster Heads (CHs) Corresponding to Fig. 4

No. of Cluster Heads	Service Rate (K-means Clustering Algorithm)	Service Rate (ACO Algorithm)
10	371.1397	644.75827
20	734.84904	963.0289
30	946.64061	1501.5987
40	1254.2681	2209.5784
50	1696.58578	3338.42747

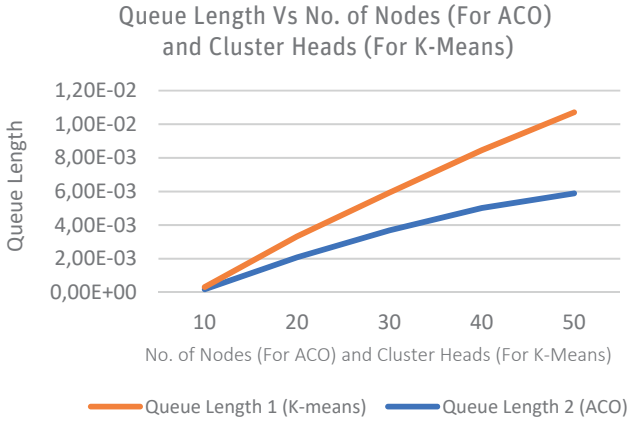


Fig. 5. Batch of Job's Queue Length for a fixed job size and varying Nodes and Cluster Heads (CHs).

TABLE III. Queue Length for a Fixed Job Size and Varying Nodes and Cluster Heads (CHs) Corresponding to Fig. 5

No. of Cluster Heads	Queue Length (ACO Algorithm)	Queue Length (K-means Clustering Algorithm)
10	0.000131	0.00018593
20	0.00125	0.00208
30	0.00225	0.00368118
40	0.003433	0.00502
50	0.00482	0.00589

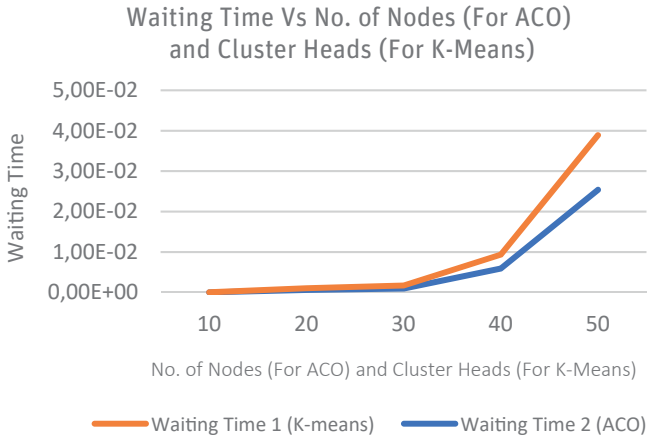


Fig. 6. Batch of Job's Waiting Time for a fixed job size and varying Nodes and Cluster Heads (CHs).

TABLE IV. Waiting Time for a fix Job Size and Varying the Nodes and Cluster Heads (CHs) Corresponding to Fig. 6

No. of Cluster Heads	Waiting Time (ACO Algorithm)	Waiting Time (K-means Clustering Algorithm)
10	0.0000122	0.0000145
20	0.000414	0.000614
30	0.000745	0.000962
40	0.00342	0.00592
50	0.0135	0.0254

Fig.7 represents the effect of both ACO and K-means clustering algorithm Response time when we vary the job size from *200* to *1000* for a fixed number of Nodes (For ACO) and CHs (For K-means) that is *10,* with the numeric data represented in Table V. We observed that the response time of ACO is again better than the K-means Clustering Algorithm.

Fig. 7. Batch of Job's Response Time for varying No. of jobs and fixed Nodes and Cluster Heads (CHs).

TABLE V. Response Time for No. of Jobs and Fixed Nodes and Cluster Heads (CHs) Corresponding to Fig. 7

No. of Jobs	Response Time (ACO)	Response Time (K-means Clustering Algorithm)
200	0.2714044	0.4507078
400	0.86546546	0.938945
600	1.826384	2.1509484
800	2.5126439	3.2269072
1000	3.293245	5.0905315

Fig. 8. Batch of Job's Service Rate for varying No. of jobs and fixed Nodes and Cluster Heads (CHs).

TABLE VI. Service Rate for varying No. of Jobs and Fixed Nodes and Cluster Heads (CHs) Corresponding to Fig.8

No. of Jobs	Service Rate (K-means Clustering Algorithm)	Service Rate (ACO Algorithm)
200	224.4072	616.66515
400	479.8531	913.67953
600	714.88793	1480.2342
800	1046.12343	2279.5486
1000	1552.5966	3425.1479

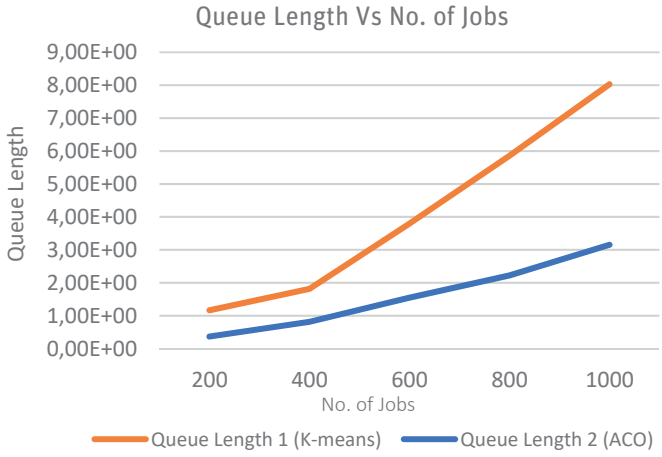


Fig. 9. Batch of Job's Queue Length for varying No. of Jobs and fixed Nodes and Cluster Heads (CHs).

TABLE VII. Queue Length for Varying No. of Jobs and Fixed Nodes and Cluster Heads (CHs) Corresponding to Fig. 9

No. of Jobs	Queue Length (ACO Algorithm)	Queue Length (K-means Clustering Algorithm)
200	0.36539	0.800113
400	0.815	1.00149
600	1.550283	2.24586
800	2.223213	3.63284
1000	3.15293	4.876906

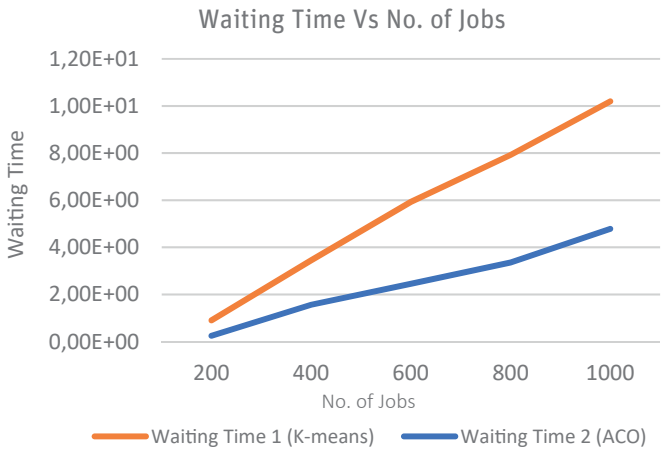


Fig.10. Batch of Job's Waiting Time for varying No. of jobs and fixed Nodes and Cluster Heads (CHs).

TABLE VIII. Waiting Time for Varying No. of Jobs and Fixed Nodes and Cluster Heads (CHs) Corresponding to Fig. 10

No. of Jobs	Waiting Time (ACO)	Waiting Time (K-means Clustering Algorithm)
200	0.2474654	0.654986
400	1.56216	1.89234
600	2.45559107	3.478343
800	3.354562	4.564951
1000	4.784304	5.4124101

VI. Conclusion

IoT is poised to change the way of living. With huge heterogeneity and dynamicity in IoT, the response time should be ensured to be as low as possible for better network performance leading to an efficient and smart IoT. The response time has effect on the use of energy cost per job for the system [25]. The optimization of the response time for transmission of data/jobs in the entire system of IoT environment automatically optimizes the cost/use of energy. This work contains a comparative analysis between ACO and K-means algorithms based on a job/message scheduling model for IoT, based on an *N* layered network. Performance measures such as the average queue length, average waiting time and average response time have been derived, plotted and analyzed. It is noticed that the ACO offers better performance for the considered parameters. The model gains significance due to the fact that efficient message forwarding in IoT will ensure the optimum use of sensor energy to realize a truly smart framework.

References

[1] O. Said, "Analysis, design and simulation of Internet of Things routing algorithm based on ant colony optimization", International Journal of Communication Systems, Wiley, 2016.

[2] S. Kumar, Z. Raza, "Internet of Things: Possibilities and Challenges", International Journal of Systems and Service-Oriented Engineering (IJSSOE), Vol.7, no.3, pp. 32-52, July-September 2017 (ISSN 1947-3052).

[3] K. Kumar, S. Kumar, O. Kaiwartya, Y. Cao, J. Lloret, N. Aslam, "Cross-Layer Energy Optimization for IoTEnvironments: Technical Advances and Opportunities", Energies, 2017.

[4] S. Kumar, Z. Raza, "Using Clustering Approaches for Response Time Aware Job Scheduling Model for Internet of Things(IoT)", International Journal of Information Technology, Springer, Vol. 9, no. 2, pp. 177-195, June 2017.

[5] S Vimal, M Khari, N Dey, RG Crespo, YH Robinson, "Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT", Computer Communications 151, pp. 355-364, 2020

[6] E. Dave, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything", Cisco Internet Business Solutions Group, pp. 1-11, April 2011.

[7] Dr. Vermesanovidiu, Dr. F. Peter, G.Patrick, G.Sergio, S. Harald, Dr. B. Alessandro, J. Ignacio, Dr. M. Margaretha, Dr. H. Mark, Dr. E. Markus, Dr. D. Pat, "Internet of Things- Global Technological and Societal Trends", published by: River Publishers, ISBN: 9788792329677.

[8] A. C. Charu, A.Naveen, S. Amit, "The Internet of Things: A Survey from the Data-Centric Perspective", Springer Science+Business Media New York, 2013, pp 383-428, ISBN 978-1-4614-6309-2.

[9] A. Saima, Y. Kun "A QoS Aware Message Scheduling Algorithm in Internet of Things Environment", IEEE Online Conference on Green Communications (Online Green Comm), 2013.

[10] Y. Lu, W. Hu, Study on the Application of Ant Colony Algorithm in the Route of Internet of Things", International Journal of Smart Home, Volume 7, No. 3, May, 2013.

[11] S. Saatchi, C. C. Hung, "Hybridization of the Ant Colony Optimization with the K-Means Algorithm for Clustering", Springer, pp. 511 – 520, 2005.

[12] C.I. Mary, DR. S.V. K. Raja, "Refinement of Clusters from k-means with Ant Colony Optimization", Journal of Theoretical and Applied Information Technology, 2009.

[13] M. Khari, et al., "Performance analysis of six meta-heuristic algorithms over automated test suite generation for path coverage-based optimization", Soft Computing, In Press, 2019

[14] J. Lu, R. Hu, "A new hybrid clustering algorithm based on K-means and ant colony algorithm", Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).

[15] C. Cheng, Z. Qian, "An IoT Ant Colony Foraging Routing Algorithm Based on Markov Decision Model", International Conference on Soft Computing in Information Communication Technology (SCICT 2014).

[16] M. Dorigo, G. D. Caro, "Ant Colony Optimization: A New Meta-Heuristic", IEEE, 1999.

[17] D. Merkle, M. Middendorf, H. Schmeck, "Ant Colony Optimization for Resource-Constrained Project Scheduling", IEEE, Transactions on Evolutionary Computation, Volume 6, No. 4, August, 2002.

[18] M. Frey, M. Günes, "Attack of the Ants: Studying Ant Routing Algorithms inSimulation and Wireless Testbeds", arXiv:1409.0988v1 [cs.NI] 3 Sep 2014.

[19] M.Głabowski, B.Musznicki, P. Nowak, P. Zwierzykowski, "Shortest Path Problem Solving Based on Ant Colony Optimization Metaheuristic", Image Processing & Communication, Volume 17, No. 1-2, pp. 7-18, 2017.

[20] P. Yuqing, H. Xiangdan, L. Shang, "The K-means Clustering Algorithm Based on Density and Ant Colony", IEEE Int. Conf. Neural Networks & Signal Processing, Nanjing, China, December 14-17, 2003.

[21] B. Mamalis, D. Gavalas, C. Konstantopoulos, G. Pantziou, "Clustering in Wireless Sensor Networks," Zhang/RFID and Sensor Networks AU7777. Proof Page 323 2009-6-24, 2012.

[22] R.Jain, "Introduction to Queueing. In: The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling," ch. 30. John Wiley & Sons, Inc., New York, 1991.

[23] Trivedi, K. S., "Probability and Statistics with Reliability, Queuing and Computer Science Applications", Prentice Hall, 1982.

[24] R. L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G.R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey, Annals of Discrete Mathematics", Volume 5, pp 287–326, 1979.

[25] E. Gelenbe, R. Lent, "Optimising Server Energy Consumption and Response Time", Theoretical and Applied Informatics, Volume 24, Issue, 4, pp 257-270, November 2012, ISSN: 1896-5334.

[26] S. Kumar, Z. Raza, "A Priority Based Message Response Time Aware Job Scheduling Model for the Internet of Things (IoT)", International Journal of Cyber Physical System (IJCPS), IGI Global, Volume 1, Issue 1, pp 1-14, January-June 2019, ISSN: 2577-4867, DOI: 10.4018/IJCPS.2019010101.

[27] C. G. García, E. R. N. Valdez, V. G. Díaz, C. P. G.Bustelo, J. M. C. Lovelle, "A Review of Artificial Intelligence in the Internet of Things, International Journal of Interactive Multimedia and Artificial Intelligence, Volume 5, Issue 4, pp 9-20, 2019.

[28] C. G. García, D. M. Llorián, J. M. C. Lovelle, "A Review about Smart Objects, Sensors and Actuators", International Journal of Interactive Multimedia & Artificial Intelligence, Volume 4, Issue 3, 2017.

[29] J. Molano, J. Lovelle, C. Montenegro, J. Granados, R. Crespo, "Metamodel for Integration of Internet of Things, Social Networks, the Cloud and Industry 4.0", Journal of Ambient Intelligence and Humanized Computing, Volume 9, Issue 3, pp 709-723, 2018.

[30] S. Kumar, Z. Raza, "Using Supply Chain Management Approach for Message Forwarding for Internet of Things (IoT)", International Conference on Technology, Engineering and Science (IConTES), Volume 4, pp 21-27, 2018.

[31] S. Kumar, Z. Raza, "A K-Means Clustering Based Message Forwarding Model for Internet of Things (IoT)", International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, pp 604-609, 2018.

[32] V. García-Díaz et al., "TALISMAN MDE framework: an architecture for intelligent model-driven engineering", International Work-Conference on Artificial Neural Networks (IWANN), Springer, pp. 299-306, 2009.

Sumit Kumar

Sumit Kumar is currently an Assistant Professor in the Department of Information Technology, Gopal Narayan Singh University, Sasaram, Bihar, India. He received M. Tech. and PhD degrees in computer science from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. He received MCA degrees in Ranchi University, Ranchi, India. His research interest is Internet of Things (IoT).

Vijender Kumar Solanki

Vijender Kumar Solanki, Ph.D., is an Associate Professor in Department of Computer Science & Engineering, CMR Institute of Technology (Autonomous), Hyderabad, TS, India. He has more than 11 years of academic experience in network security, IoT, Big Data, Smart City and IT. Prior to his current role, he was associated with Apeejay Institute of Technology, Greater Noida, UP, KSRCE (Autonomous) Institution, Tamilnadu, India & Institute of Technology & Science, Ghaziabad, UP, India. He has attended an orientation program at UGC-Academic Staff College, University of Kerala, Thiruvananthapuram, Kerala &Refresher course at Indian Institute of Information Technology, Allahabad, UP, India. He has authored or co-authored more than 50 research articles that are published in journals, books and conference proceedings. He has edited or co-edited 10 books in the area of Information Technology. He teaches graduate & post graduate level courses in IT at ITS. He received Ph.D in Computer Science and Engineering from Anna University, Chennai, India in 2017 and ME, MCA from Maharishi Dayanand University, Rohtak, Haryana, India in 2007 and 2004, respectively and a bachelor's degree in Science from JLN Government College, Faridabad Haryana, India in 2001.He is Editor in International Journal of Machine Learning and Networked Collaborative Engineering (IJMLNCE) ISSN 2581-3242, Associate Editor in International Journal of Information Retrieval Research (IJIRR), IGI-GLOBAL, USA,ISSN: 2155-6377 | E-ISSN: 2155-6385 also serving editorial board members with many reputed journals. He has guest edited many volumes, with IGI-Global, USA, InderScience& Many more reputed publishers.

Saket Kumar Choudhary

Saket Kumar Choudhary is an assistant professor in FCA, MRIIRS, Faridabad, Harayana, India. He has obtained his master degrees in Mathematics from the University of Allahabad, Allahabad, India in 2005, Master of Computer Application (MCA) from UPTU, Lucknow, India in 2010, Master of Technology (M.Tech) from Jawaharlal Nehru University, New Delhi, India in 2014. He is Ph.D (Computer Science and Technology) School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. Currently, he is working as an assistant professor in MRIIRS, Faridabad. His research interest includes mathematical modeling and simulation, dynamical systems, computational neuroscience: modeling of single and coupled neurons, computer vision, digital image processing, machine learning and artificial intelligence.

Ali Selamat

Ali Selamat has received a B.Sc. (Hons.) in IT from Teesside University, U.K. and M.Sc. in Distributed Multimedia Interactive Systems from Lancaster University, U.K. in 1997 and 1998, respectively. He has received a Dr. Eng. degree from Osaka Prefecture University, Japan in 2003. Currently, he is the Dean of Research Alliance in Knowledge Economy (K-Economy RA) UTM. He is also a professor at the Software Engineering Department, Faculty of Computing UTM. Previously he was an IT Manager at School of Graduate Studies (SPS), UTM. He is also a head of Software Engineering Research Group (SERG), K-Economy Research Alliance, UTM. He is the editors of International Journal of Digital Content Technology and its Applications (JDCTA), International Journal of Advancements in Computing Technology (IJACT) and International Journal of Intelligent, Information and Database Systems (IJIIDS). His research interests include software engineering, software agents, web engineering, information retrievals, pattern recognitions, genetic algorithms, neural networks and soft-computing.

Rubén González Crespo

Dr. Rubén González Crespo has received a PhD in Computer Science Engineering. He is also Industrial Engineering and MSc in Computer Science and Project Management. Currently he is Vice Rector of Academic Affairs from UNIR. He is advisory board member for the Ministry of Education at Colombia and evaluator from the National Agency for Quality Evaluation and Accreditation of Spain (ANECA). He is president from Engineering and Architecture Commission in Fundación Madrid+D. He is patron of the Spain Free Software Foundation. He is member of different committees at ISO Organization. He has published more than 200 research works in several relevant journal. He also manages different research projects in IoT, TEL, and Artificial Intelligence.

Multilayer Feedforward Neural Network for Internet Traffic Classification

N. Manju[1]*, B. S. Harish[2]*, N. Nagadarshan[1]

[1] Department of Information Science and Engineering, Sri Jayachamarajendra College of Engineering, Mysuru (India)
[2] Department of Information Science and Engineering, JSS Science and Technology University, Mysuru (India)

Received 1 April 2019 | Accepted 4 November 2019 | Published 25 November 2019

Abstract

Recently, the efficient internet traffic classification has gained attention in order to improve service quality in IP networks. But the problem with the existing solutions is to handle the imbalanced dataset which has high uneven distribution of flows between the classes. In this paper, we propose a multilayer feedforward neural network architecture to handle the high imbalanced dataset. In the proposed model, we used a variation of multilayer perceptron with 4 hidden layers (called as mountain mirror networks) which does the feature transformation effectively. To check the efficacy of the proposed model, we used Cambridge dataset which consists of 248 features spread across 10 classes. Experimentation is carried out for two variants of the same dataset which is a standard one and a derived subset. The proposed model achieved an accuracy of 99.08% for highly imbalanced dataset (standard).

Keywords

Classification, Feature Transformation, Feedforward Network, Internet Traffic.

DOI: 10.9781/ijimai.2019.11.002

I. Introduction

RECENTLY, the internet traffic classification has been frequently used in order to improve service quality in enterprise networks, developing new service packets, sharing the bandwidth source and performing traffic analysis effectively. There are various methods of internet traffic classification such as port based, payload based and flow statistical based approach. Port based classification is very simple but produces inaccurate results because of dynamic assignment of protocols [1], [2]. Payload based method have been proposed and presented by many researchers, which analyzes packet content to identify the internet traffic. Payload method is popular but has few limitations like lawful inspection, encrypted data cannot be handled and so on. To overcome these issues, a flow statistical based approach is used for IP traffic classification [3], [4]. The above said classification problems can be solved using machine learning approaches.

Machine learning consists of supervised learning and unsupervised learning. In unsupervised learning, class labels are unknown and a common technique of this type is called as clustering. In clustering, identical features are grouped as cluster. On the other hand, the classification can be performed using supervised learning. Supervised learning tries to figure out classes from the training data, and test data have been measured by studying the learning data [5]. The most used data discriminators in flow based method is given in [6] for the assessment of the classification task.

* Corresponding author.
E-mail addresses: manjun007@sjce.ac.in (N. Manju), bsharish@jssstuniv.in (B. S. Harish).

The paper is organized as follows: Section II presents the literature, section III depicts the proposed model, section IV discusses the results and section V presents the conclusion with future work.

II. Related Work

A lot of network applications run on the Internet and generate a large amount of traffic every day. It is a daunting task to manage the huge and mixture traffic. Network traffic classification is more and more important along with the development of Internet.

The basic variants of machine learning approaches are supervised and unsupervised learning. Unsupervised method contains only the desired input and there are no output labels. A common unsupervised method is known as clustering [7], [8]. The supervised method is applied to analyze the learning data. Supervised learning method has both desired inputs and desired outputs. One of the initial important works in this area is the study in [9] which classifies the network traffic by applying the supervised naive bayes method on the flow statistical features. Classification of internet traffic is done by analyzing a packet via the capturing tool and determining the 248 features in [10]. Later, traffic classification is done using support vector machine with an accuracy of 90% on various types of internet traffic datasets [11]. Bayesian neural networks is applied for Cambridge dataset and achieved a class wise accuracy using 2441 flows for 7 classes ranging from 91% to 99.9% of accuracy [12]. Various classification algorithms are compared and evaluated in [13]. Such classification algorithms are: Naïve Bayes using both Naïve Bayes Descritization (NBD) and Kernel density estimation (NBK), C4.5 decision tree, Bayesian Network (BayesNet) and Naïve Bayes Tree (NBTree). Experimentation result shows that the C4.5 identifies the network flows faster than any other

four algorithms with respect to computational performance. Further, other studies reported using the machine learning algorithms Support Vector Machine (SVM) and Decision Tree. The result shows that; decision tree is computationally efficient compared to SVM as seen in [14]. Identification of Voice over Internet Protocol (VoIP) using machine learning approaches such as C5.0, AdaBoost and Genetic Programming is presented in [15]. Result shows that, C5.0 performs better compared to other two algorithms.

Artificial Immune System (AIS) is proposed and used to classify the malicious packets in Intrusion Detection System [16]. Inspired by this approach, AIS based algorithms are proposed to classify the flow of internet traffic and the performance of the proposed method is evaluated with and without using kernel function [17]. The size of the dataset used is 1000 flows. The range of the accuracy achieved is between 81.8% and 91% without using kernel function. Using linear kernel, the performance achieved was 92.3% of accuracy. Other two algorithms such as Naïve Bayes and SVM achieved an accuracy of 82.2% and 44.10% respectively [18]. Imbalance nature of the dataset used for the experimentation affects the algorithm proposed for the few classes as stated in [19]. Data Gravitation based Classification (DGC) model is proposed in [20] and 12 UCI datasets were used for the experimentation. DGC achieves better accuracy. However, it achieved less accuracy for imbalanced data sets. Hence, DGC suffers in using the unbalanced datasets. To address this problem, Imbalanced DGC (IDGC) model is proposed and Amplified Gravitation Coefficient (AGC) is used to compute the gravitation [21]. Based on the results of the proposed models, it can be concluded that most of the standard machine learning classification models are not much effective with imbalanced internet traffic data [22]. To enhance the performance of imbalance classification, Under-sampling DGC is proposed in [23]. The proposed method generates a set of training groups for re-balancing. Generated new training set are selected and employed to DGC model to get better classification accuracy. Later, voting policy is used to obtain the classification model. Therefore, it gives promising result for the imbalanced dataset. The proposed method is applied on 22 less imbalanced and 22 high imbalanced datasets. Class Oriented Feature Selection (COFS) is proposed to improve the recall and precision for minority classes maintaining overall high accuracy in [24]. Cost sensitive learning is studied for multiclass imbalance problem in [25] and misclassification cost with respect to dependent class is investigated. Weighted Cost Matrix (WCM) is proposed and compared with Flow rate based Cost Matrix (FCM). The obtained result gives better accuracy for the proposed method.

A. Motivation

The work is carried out for two reasons such as, a) To build multi layer neural network and b) to propose a model which can handle the class imbalance problem. Authors in [26] has proposed Extreme Learning Machine (ELM) which is used to classify the internet traffic. Extreme Learning Machine is a fast learner neural network model which randomly takes hidden layer weights. It was proposed for the training of single hidden layer feedforward neural networks. The weight and threshold values in traditional single hidden layer feedforward neural networks should be renewed by gradient-based learning algorithms. To provide a good solution, this situation may cause both an extension of the learning process and the error may be encountered in a local point. The change can be made in the momentum value for the problem of errors being encountered in a local point. However the change will not make a contribution to the learning process. The weights and thresholds in a single hidden layer feedforward network do not affect the performance of the network.

As the application of neural networks is increasing day to day, we exploited the application of multilayer feed-forward neural network to solve the problem of Internet traffic classification. The proposed network is used towards the basic deep neural networks and thus we used multiple layers in our architecture. The main objective of using the architecture of multi layer feedforward neural network is because of the capability of handling large amount of datasets. Multilayer feedforward has the ability to detect complex nonlinear relationships between dependent and independent variables. Further, Multilayer feedforward exhibits the various advantages like adaptability in learning, non-linearity property, mapping between input and output, robustness in accommodating the noise. Using machine learning with feedforward neural network with multiple layers and with other parameters would be able to construct the deep learning architecture in near future.

III. Proposed Methodology

In this model, we are proposing a multi layer feedforward neural network for internet traffic classification. The main objective of using this network is that, it has high learning rate on non linear data [27]. It is quite obvious that, all the attributes of the dataset description are not same. Hence, it is highly difficult to classify without normalizing the data. Therefore, we first normalize the data before sending it to the proposed model. Normalizing is done using the eq. (1):

$$f(x) = (x - \mu) / \sigma \tag{1}$$

Where,

x is the column vector (features).

μ is the mean of x.

σ is the standard deviation of x.

After normalizing the features, the new features will be in the range of [-1, 1]. This helps us to classify the data without having bias towards the high biased features.

The proposed methodology uses multilayer feedforward neural network with six layers. Out of six layers, four are hidden layers, one is input layer, and one is output layer. Finally we use the softmax activation function for classification.

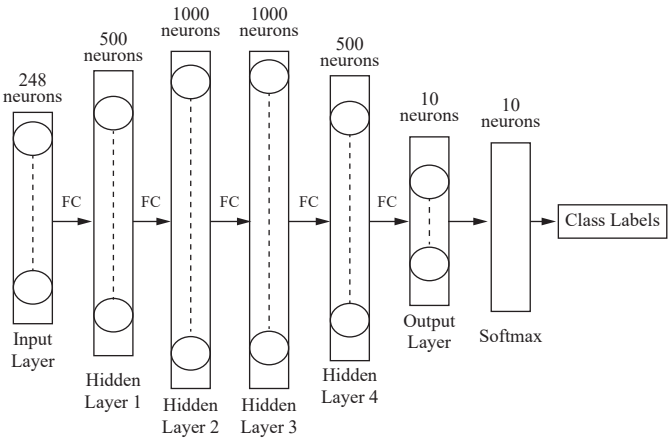


Fig.1.Proposed Multi Layer Feedforward Neural Network Model (FC refers to Fully Connected).

As we can see from Fig.1, the input layer consists of 248 neurons because of 248 features. The first hidden layer consists of 500 hidden neurons, the second and third hidden layers consist of 1000 hidden neurons each. Fourth and the final hidden layer consist of 500 hidden neurons. The final output layer consists of 10 neurons as there are 10 classes. At last we use softmax as the loss function.

Assuming the preprocessed features are encoded, we first extract the features and make it 1000 dimensions by decoding the features. After the model understands the data deeply, it is again transformed to the lower dimensions to get only the required features by encoding. This process can be understood by the loss functions. In this procedure, all the hidden layers can be viewed as feature transformational units.

The first layer as mentioned earlier is known as the input layer whose number of neurons are always equal to the number of attributes in the dataset. The dataset consists 248 attributes.

The next layer is the first hidden layer. In this layer, all the neurons of the previous layers are connected to all the neurons of the next layer through dot product of the weights (eq. (2)).

$$f(\mathrm{x}) = wx + b \tag{2}$$

where,

w is the weights of the current layer

x is the output of the previous layer (here it the input features)

b is the bias.

Initially weights are initialized with normal Gaussian values. This type of weight initialization is used across various neural network descendants like Convolution Neural Network, Recurrent Neural Network, etc. The next layer i.e., the first hidden layer consists of 500 neurons. To achieve better performance, the number of hidden layers in multi layer perceptron should be as twice the number of neurons which takes the advantage mentioned in [28]. Therefore, we double and round off the number of input features to get 500 neurons. Further, we apply eq. (2) and we pass it to the activation function or kernel. The authors in [26] used wavelet kernel as their activation function. The proposed model uses Rectified Linear Unit (ReLU shown in eq. (3) as the activation function as it is good in capturing data with moderate and large number of features [29]. Further it also helps in faster convergence and it is by far simpler than wavelet kernel.

$$f(\mathrm{x}) = \max(0, \mathrm{X}) \tag{3}$$

where,

x is the input to the ReLU function

Fig. 2 shows the graph of the ReLU activation function, which squashes every negative input to zero and keeps every positive input as the same.

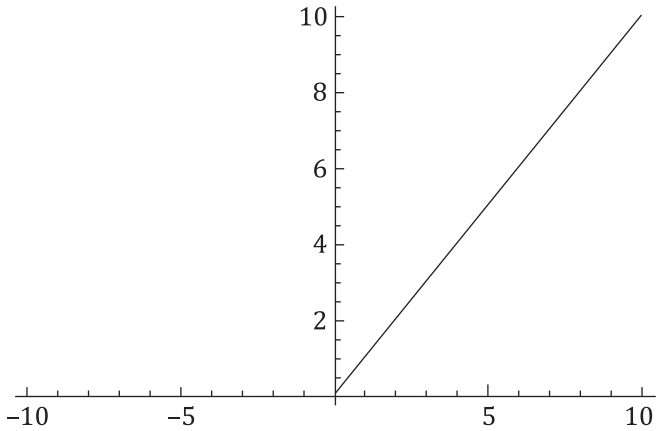


Fig. 2. Graph of the ReLU activation function.

The next layer, i.e., second hidden layer consists of 1000 neurons as it is still needed to extract the features from the data that are going to double the number of neurons from the previous layer. In particular, this layer helps in understanding the data much deeper. In this work, ReLU is used as the activation function. The following layer is the same as previous layer, which also contains 1000 neurons and uses ReLU as the activation function.

In the third hidden layer, we tend to project it to the lower dimensional to reduce the number of neurons and hence it contains 500 neurons. It also uses ReLU as the activation function. The main objective of doing this is to extract only the relevant features.

The last layer is the output which contains 10 neurons as there are 10 classes. After the output layer we have the softmax layer which is used as the loss function and is given by:

$$S(\mathrm{y}_i) = \frac{e^{y_i}}{\sum_j e^{y_i}} \tag{4}$$

where,

y_i is the score of class j

e is the standard mathematical constant ≃ 2.718

The softmax function is chosen among other loss functions because the range of the softmax is [0, 1] which does not require more space and the sum of all the scores is equal 1. Hence it is a probabilistic score and losses can be predicted relatively.

The back propagation algorithm is used to back propagate through the network updating the weights through adam optimizer [30]. It is used for optimization which automatically adapts and updates its parameters. This model is executed for 1000 epochs with a batch size of 2486. The increase in number of epochs leads to the over fitting problem. Therefore epoch was set to 1000.

IV. Experimental Section

A. Experimental Setup

The experiment was carried on a machine with the operating system ubuntu version 16.04 with 12 GB RAM and the secondary storage capacity of 1TB. The processor is Intel Core i7-7700HQ CPU at 2.80GHz × 8. Implementation is done using a 64 bit machine.

B. Dataset

The dataset used is a publicly available Cambridge dataset which consists of 248 features. The total number of classes are 10. Two variants of the same dataset are used. One standard dataset is highly imbalanced and the other is less imbalanced, which is a derived dataset. In the standard dataset, flows are more biased towards WWW class and least biased towards multimedia class. The dataset is divided in the ratio of 70% for training and 30% for testing. Table I shows the characteristics of datasets used for experimentation.

C. Results and Discussions

In the following section, we discuss the results obtained on the dataset01 which is the high imbalanced dataset. The main objective of this experimentation is to overcome the problem of imbalance dataset which is one of the major challenges in the research. The proposed model avoids the bias result and gives the better result.

As we can see from Table II, which contains the confusion matrix for imbalanced dataset, almost all the classes have been classified correctly which lead to an accuracy of 99.08%, which is around 3% greater than the accuracy achieved in [26]. Hence the proposed model achieves a greater result. We can also see that the number of misclassification samples of each class is far less than the number of samples correctly classified samples.

TABLE I. Characteristics of Datasets Used for Experimentation

Class No.	Class Name	Standard Dataset **(Dataset01)** **Flow Count**	Flow distribution Ratio in %	Derived Dataset **(Dataset02)** **Flow Count**	Flow distribution Ratio in %
1	ATTACK	122	00.49	1793	07.50
2	DATABASE	238	00.95	2648	11.08
3	FTP-CONTROL	149	00.59	3000	12.55
4	FTP-DATA	1319	05.30	3000	12.55
5	TFP-PASV	43	00.17	2688	11.24
6	MAIL	4146	16.67	3000	12.55
7	MULTIMEDIA	87	00.34	576	02.41
8	P2P	339	01.36	2094	08.76
9	SERVICES	206	00.82	2099	08.78
10	WWW	18211	73.25	3000	12.55
Total Number of Flows		**24860**	**~100.00**	**23898**	**~100.00**

TABLE II. Confusion Matrix of 10 Classes for Dataset01

Predicted \ Actual	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	**30**	0	0	0	0	0	3	1	0	14
2	0	**68**	0	0	1	0	0	0	0	3
3	0	0	**42**	0	0	5	0	0	0	0
4	0	0	0	**369**	0	1	0	1	0	1
5	0	0	0	0	**7**	0	0	3	0	2
6	2	1	0	0	0	**1195**	0	0	0	10
7	2	0	1	0	1	0	**17**	0	0	0
8	0	0	0	0	0	0	2	**81**	0	2
9	0	0	0	0	0	0	0	0	**71**	0
10	0	2	0	0	2	3	0	5	0	**5510**

TABLE III. Table Showing, Recall, Precision and F_1 Score of All the Classes for Dataset01

Class	Recall	Precision	F_1 Score
1	0.88	0.62	0.73
2	0.95	0.94	0.95
3	0.97	0.89	0.93
4	1.00	0.99	0.99
5	0.63	0.58	0.60
6	0.99	0.98	0.99
7	0.77	0.80	0.79
8	0.89	0.95	0.92
9	1.00	1.00	1.00
10	0.99	0.99	0.99
Average	0.91	0.87	0.89

The table III shows that we have compelling precision and recall values. As we can see the average precision is of 0.87, which is quite high. The average recall value is 0.91 that is also very high, which shows low false-negative rate. The F_1 score of 0.89 shows a good result for a dataset01 shown in Table I. The Precision, Recall and F_1 score show that the model is successful in handling even the less balanced data.

Further, we discuss the results achieved using the dataset02 which is a less imbalanced (derived) dataset.

TABLE IV. Confusion Matrix of 10 Classes for Dataset02

Predicted \ Actual	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	**472**	3	2	2	0	7	4	9	1	22
2	0	**780**	1	0	2	0	0	1	0	0
3	1	0	**890**	0	0	12	1	0	1	0
4	1	0	0	**907**	1	6	0	0	0	1
5	1	1	1	1	**774**	0	0	4	0	2
6	4	1	5	2	1	**908**	0	0	1	5
7	9	0	0	3	2	3	**160**	2	0	0
8	12	5	0	0	7	7	8	**593**	0	2
9	0	3	1	0	0	5	0	1	**637**	1
10	6	4	3	0	4	0	0	0	2	**852**

As we can see from Table IV, the confusion matrix for dataset02, almost all the classes have been classified correctly, which leads to an accuracy of 97.25%, which is again much greater than existing in [26].

TABLE V. Recall, Precision and F_1 Score of All the Classes for Dataset02

Class	Recall	Precision	F_1 Score
1	0.93	0.90	0.91
2	0.97	0.99	0.98
3	0.98	0.98	0.98
4	0.99	0.99	0.99
5	0.97	0.98	0.98
6	0.95	0.97	0.96
7	0.92	0.89	0.90
8	0.97	0.93	0.95
9	0.99	0.98	0.98
10	0.96	0.97	0.97
Average	0.96	0.95	0.96

Table V presents the result of Precision, Recall and F_1 Score. From the results, we can observe that the average Precision is of 0.95, which is quite high for imbalanced dataset. The average Recall value of 0.96 is also very high, which shows low false-negative rate. The F_1 score of 0.96 shows a good result for a highly balanced dataset.

TABLE VI. Accuracy Comparison Table

Methodology	Activation Function	Number of hyden layers	Accuracy in %
ELM [26]	Sigmoid	1	79.16
ELM [26]	Tangent Sigmoid	1	92.11
ELM [26]	Radial basis	1	87.30
ELM [26]	Wavelet	1	96.57
Standard dataset (Dataset01)	**ReLU**	**4**	**99.08**
Derived dataset (Dataset02)	**ReLU**	**4**	**97.25**

Table VI shows that ELM neural network with Sigmoid kernel and using one hidden layer achieved an accuracy of 79.16% and ELM with Tangent Sigmoid kernel achieved 92.11% of accuracy. ELM with Radial basis attained 87.30% accuracy and ELM with Wavelet kernel achieved 96.57% of accuracy. Further, the proposed model out performed with an accuracy of 99.08% for dataset01 and 97.25% accuracy for dataset02, which is more than the accuracy reported in [26].

The increase in accuracy is due to the design architecture of neural network model discussed in section III. The model first extracts the number of features and projects it to the higher dimensions. Later, the model understood the data deeply and it is transformed to the lower dimensions to get only the required features. Hence it is a fact that the proposed classifier has successfully solved the problem of internet traffic classification for both less imbalanced and highly imbalanced dataset. Even though the model is intended for high imbalanced dataset, table IV, V and VI justifies that it also performed well for a less imbalanced dataset.

V. Conclusion

Currently, internet traffic classification is becoming popular due to growth in the internet usage and utility applications. However, network management is a key challenge to the service providers. Efficient network management leads to provide quality of service, ensure security and high computing performance of the network. Another issue with the internet traffic is about handling imbalance dataset. Hence there is a need of addressing the existing problem, i.e internet data that comes with imbalance in particular. Thus, to solve this issues, we proposed a multi layer feedforward neural network model with four hidden layers (mountain mirror network). The effective transformation of the dimensions of the features helps in accurate classification and thereby increasing the accuracy. In future, we can also try with various kernels to compare the results achieved. Further, we are also intending to use dimensionality reduction technique to reduce the higher dimensional internet traffic data to lower dimensions. To test the scalability of our proposed method, we also plan to work on datasets of massive traffic in real networks.

References

[1] Kim, H., Claffy, K. C., Fomenkov, M., Barman, D., Faloutsos, M., and Lee, K., "Internet traffic classification demystified: myths, caveats, and the best practices," Proceedings of the ACM CoNEXT conference, ACM, 2008.

[2] Callado, A., Kamienski, C., Szabó, G., Gero, B. P., Kelner, J., Fernandes, S., and Sadok, D., "A survey on internet traffic identification," IEEE Communications Surveys & Tutorials, vol. 11, no. 3, pp. 37-52, 2009.

[3] Finsterbusch, M., Richter, C., Rocha, E., Muller, J. A., and Hanssgen, K.,"A survey of payload-based traffic classification approaches," IEEE Communications Surveys & Tutorials, vol. 16, no. 2, pp. 1135-1156, 2014.

[4] Ducange, P., Mannarà, G., Marcelloni, F., Pecori, R., and Vecchio, M., "A novel approach for internet traffic classification based on multi-objective evolutionary fuzzy classifiers," IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-6, July. 2017.

[5] Namdev, N., Agrawal, S., and Silkari, S.," Recent advancement in machine learning based internet traffic classification," Procedia Computer Science, vol. 60, pp.784-791, 2015.

[6] Moore, A., Zuev, D., and Crogan, M., "Discriminators for use in flow-based classification," 2013.

[7] J. Erman, A. Mahanti, M. and Arlitt, C. Williamson, "Identifying and discriminating between web and peer-to-peer traffic in the network core," Proceedings of the 16th International Conference on World Wide Web, ACM, pp. 883–892, 2007.

[8] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data, ACM, pp. 281–286, 2006.

[9] A.W. Moore, and D. Zuev, "Internet traffic classification using bayesian analysis techniques," ACM SIGMETRICS Performance Evaluation Review, vol. 33, no. 1, ACM, pp. 50–60, 2005.

[10] A. Moore, D. Zuev, and M. Crogan, "Discriminators for Use in Flow-Based Classification," Queen Mary and Westfield College, Department of Computer Science, 2005.

[11] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for TCP traffic classification," Computer Networks, vol. 53, no. 14, pp. 2476–2490, 2009.

[12] T. Auld, A.W. Moore, and S.F. Gull, "Bayesian neural networks for internet traffic classification," IEEE Transaction on Neural Networks, vol. 18, no. 1, pp. 223–239, 2007.

[13] N. Williams, S. Zander, and G.A. Armitage, "Preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," ACM SIGCOMM Computer. Communication. Review, vol. 36, no. 5, pp. 5–16, 2006.

[14] Valenti, S., Rossi, D., Dainotti, A., Pescapè, A., Finamore, A., and Mellia, M., "Reviewing traffic classification," In Data Traffic Monitoring and Analysis, pp. 123-147, Springer, Berlin, Heidelberg, 2013.

[15] Alshammari, R., & Zincir-Heywood, A. N., "Identification of VoIP encrypted traffic using a machine learning approach," Journal of King Saud University-Computer and Information Sciences, vol. 27, no. 1, pp. 77-92, 2015.

[16] Kim, J., Bentley, P. J., Aickelin, U., Greensmith, J., Tedesco, G., and Twycross, J., "Immune system approaches to intrusion detection–a review," Natural computing, vol. 6, no. 4, pp. 413-466, 2007.

[17] Schmidt, B., Kountanis, D., and Al-Fuqaha, A., "Artificial immune system inspired algorithm for flow-based internet traffic classification," IEEE 6th International Conference onIn Cloud Computing Technology and Science (CloudCom), pp. 664-667, December. 2014.

[18] Schmidt, B., Kountanis, D., and Al-Fuqaha, A. "A Biologically-Inspired Approach to Network Traffic Classification for Resource-Constrained Systems," In Proceedings of the 2014 IEEE/ACM International Symposium on Big Data Computing IEEE Computer Society, pp. 113-118, December. 2014.

[19] Schmidt, B., Al-Fuqaha, A., Gupta, A., and Kountanis, D., "Optimizing an artificial immune system algorithm in support of flow-Based internet traffic classification," Applied Soft Computing, vol. 54, pp. 1-22, 2017.

[20] Peng, L., Yang, B., Chen, Y., and Abraham, A., "Data gravitation based classification," Information Sciences, vol. 179, no. 6, pp. 809-819, 2009.

[21] Peng, L., Zhang, H., Yang, B., and Chen, Y., "A new approach for imbalanced data classification based on data gravitation," Information Sciences, vol. 288, pp. 347-373, 2014.

[22] Peng, L., Zhang, H., Chen, Y., and Yang, B, "Imbalanced traffic identification using an imbalanced data gravitation-based classification model," Computer Communications, vol. 102, pp. 177-189, 2017.

[23] Peng, L., Yang, B., Chen, Y., and Zhou, X., "An under-sampling imbalanced learning of data gravitation based classification," IEEE 12th International Conference on Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 419-425, August. 2016.

[24] Liu, Z., Wang, R., Tao, M., and Cai, X.,"A class-oriented feature selection approach for multi-class imbalanced network traffic datasets based on local and global metrics fusion," Neurocomputing, vol. 168, pp. 365-381, 2015.

[25] Zhen, L. I. U., and Qiong, L. I. U., "Studying cost-sensitive learning for multi-class imbalance in Internet traffic classification," The Journal of China Universities of Posts and Telecommunications, vol. 19, no. 6, pp. 63-72, 2012.

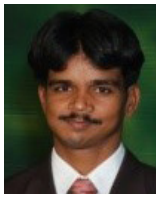
[26] Ertam, F. and Avcı, E., "A new approach for internet traffic classification: GA-WK-ELM," Measurement, vol. 95, pp.135-142, 2017.

[27] Panchal, G., Ganatra, A., Kosta, Y. P., and Panchal, D.,"Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers," International Journal of Computer Theory and Engineering, vol. 3, no. 2, pp. 332-337, 2011.

[28] Panchal, G., Ganatra, A., Kosta, Y.P. and Panchal, D., "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers," International Journal of Computer Theory and Engineering, vol. 3, no. 2, p.332, 2011.

[29] Agarap, A. F., "Deep learning using rectified linear units (relu)," arXiv preprint arXiv:1803.08375, 2018.

[30] Kingma, D. P., and Ba, J. "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

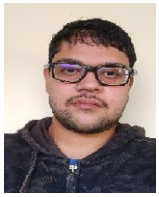


N. Manju

Obtained his B.E in Computer Science and Engineering in the year 2005 and M.Tech in Computer Network Engineering in 2008 from Visvesvaraya Technological University, Belagavi, Karnataka, India. He is presently working as an Assistant Professor in the Department of Information Science & Engineering, Sri Jayachamarajendra College of Engineering, Mysuru, Karnataka, India. He is a Life Member of ISTE. His area of interest includes Machine Learning and Computational Intelligence.

B. S. Harish

Obtained his B.E in Electronics and Communication (2002), M.Tech in Networking and Internet Engineering (2004) from Visvesvaraya Technological University, Belagavi, Karnataka, India. He completed his Ph.D. in Computer Science (2011); thesis entitled "Classification of Large Text Data" from University of Mysore. He is presently working as an Associate Professor in the Department of Information Science & Engineering, JSS Science & Technology University, Mysuru. He was invited as a Visiting Researcher to DIBRIS - Department of Informatics, Bio Engineering, Robotics and System Engineering, University of Genova, Italy from May- July 2016. He delivered various technical talks in National and International Conferences. He has been invited as a resource person to deliver various technical talks on Data Mining, Image Processing, Pattern Recognition, Soft Computing. He is also serving and served as a reviewer for National, International Conferences and Journals. He has published more than 50 International reputed peer reviewed journals and conferences proceedings. He successfully executed AICTE-RPS project which was sanctioned by AICTE, Government of India. He also served as a secretary, CSI Mysore chapter. He is a Member of IEEE (93068688), Life Member of CSI (09872), Life Member of Institute of Engineers and Life Member of ISTE. His area of interest includes Machine Learning, Text Mining and Computational Intelligence.



N. Nagadarshan

Obtained his B.E in Information Science and Engineering in the year 2019 from Sri Jayachamarajendra College of Engineering. His area of interests are Deep Learning, Biometrics and also he is very much passionate about Spiking Neural Networks. He also interned at Indian Statistical Institute, Kolkata. He is currently working on Spiking Neural Networks at Indian Institute of Science, Bengaluru.

A Multicriteria Optimization for Flight Route Networks in Large-Scale Airlines Using Intelligent Spatial Information

Mostafa Borhani*, Kamal Akbari, Aliakbar Matkan, Mohammad Tanasan

Shahid Beheshti University (Iran)

Received 30 May 2019 | Accepted 14 November 2019 | Published 25 November 2019

Abstract

Air route network optimization, one of the airspace planning challenges, effectively manages airspace resources toward increasing airspace capacity and reducing air traffic congestion. In this paper, the structure of the flight network in air transport is analyzed with a multi-objective genetic algorithm regarding Geographic Information System (GIS) which is used to optimize this Iran airlines topology to reduce the number of airways and the aggregation of passengers in aviation industries organization and also to reduce changes in airways and the travel time for travelers. The proposed model of this study is based on the combination of two topologies – point-to-point and Hub-and-spoke – with multiple goals for causing a decrease in airways and travel length per passenger and also to reach the minimum number of air stops per passenger. The proposed Multi-objective Genetic Algorithm (MOGA) is tested and assessed in data of the Iran airlines industry in 2018, as an example to real-world applications, to design Iran airline topology. MOGA is proven to be effective in general to solve a network-wide flight trajectory planning. Using the combination of point-to-point and Hub-and-spoke topologies can improve the performance of the MOGA algorithm. Based on Iran airline traffic patterns in 2018, the proposed model successfully decreased 50.8% of air routes (184 air routes) compared to the current situations while the average travel length and the average changes in routes were increased up to 13.8% (about 100 kilometers) and up to 18%, respectively. The proposed algorithm also suggests that the current air routes of Iran can be decreased up to 24.7% (89 airways) if the travel length and the number of changes increase up to 4.5% (32 kilometers) and 5%, respectively. Two intermediate airports were supposed for these experiments. The computational results show the potential benefits of the proposed model and the advantage of the algorithm. The structure of the flight network in air transport can significantly reduce operational cost while ensuring the operation safety. According to the results, this intelligent multi-object optimization model would be able to be successfully used for a precise design and efficient optimization of existing and new airline topologies.

Keywords

Multi-Objective Optimization, Genetic Algorithm, Airway Topology, GIS, Artificial Intelligence, Non-Dominated Sorting Genetic Algorithm II (NSGA-II).

DOI: 10.9781/ijimai.2019.11.001

I. Introduction

Airline network topology consists of many different and wide-ranging parameters such as travel length, the number of airways, and changes in routes, as well as some rules and constraints like re-routing, flight-level changes, and a limited airport capacity [1]. Air route network optimization as a multipurpose optimization of an indescribable type or the so-called Non-deterministic Polynomial-time hard (NP-Hard) [2] [3] [4] should be considered in hub network design problem [5]. This paper suggests employing the capabilities of Geographic Information Systems (GIS) as powerful tools regarding the management, analysis, and offer of spatial information for designing topology of airline route networks [6]. Moreover, a Non-dominated sorting genetic algorithm II (NSGA-II) [7] as a multi-purpose optimization algorithm is implemented in this study. The Multi-objective Genetic Algorithm (MOGA) is exploited because the obtained Pareto Front of MOGA is very accurate and fast [8]. MOGA as one type of well-known and robust genetic optimization algorithms is one of the most popular multicriteria optimization algorithms which is also a potent technique for solving problems concerning research and optimization in the real world. Using the Pareto ranking and the crowding distance in the ranking of the individuals encourage various applications especially, those regarding GIS to exploit this algorithm because of the absence of additional parameters in MOGA as well as its high computational speed [7].

GIS knowledge in spatial modeling and simulation especially with soft-computing methods [9] such as multi-objective genetic algorithm [10] can be used in designing airways topology. Intelligent algorithms such as multi-objective optimization algorithms could help GIS dealing with matters concerning spatial decision-making [11]. Multi-objective evolutionary algorithms take a more evolutionary approach to learning than traditional artificial intelligence (AI) [12] and it is usually facing a vast amount of data and processing. Typically, the environment that requires optimization has different objectives and complications, so, a fast algorithm could be a convenient option [13] [14].

Modelling the air transport with complex networks [15] is one of the essential issues in the transport geography and GIS, which can be optimized by methods for generating non-dominated solutions (the weighting, and constraint methods), the distance-based methods (such as compromise programming, goal programming, and reference point

* Corresponding author.
E-mail address: mo_borhani@sbu.ac.ir

methods), and interactive methods. By choosing a proper topology for air routes in GIS, airlines can save costs and reduce risk in the air industry [5]. Moreover, aviation industries could offer cheaper yet high-quality services which will enhance the travelers' satisfaction and finally boost the air industry.

There have been numerous studies about the designing of airline topology and its optimization. O'Kelly introduced a hub locating using a single-center air network that was modeled using a numerical scheduling method [2]. His results demonstrated that this matter belongs to NP-hard issues and it requires a counting method based on innovative algorithms to be solved [2]. He assumed only one hub. Park and Sohn tried to explain a two-hub locating through linear planning [2]. They concluded that when there are two hubs, we can use linear planning [16]. Marianova and Serra [17] worked on modeling hub locating in airline networks by considering swarm at intermediate airports in the presence of M/D/c queue. In their model, they used an exploratory method based on forbidden search and implemented it on American Airways Dataset (CAB) [17]. Wagner performed clustering through a genetic algorithm and banned search to solve incapacitated multiple allocation hub location problems (UHLP-M) [18]. Mohammadi et al. considered swarm at hubs once all its functionality has been lost and figured it out through a multi-server queueing Model, M/M/c queue system (or Erlang–C Model) [19]. Alumur et al. used mixed-integer programming for locating intermediate airports at multiple levels for Turkey and studied sensitivity analysis and performed a more accurate analysis of hub locating [20]. Taghipourian et al. examined a hub network in the event of impairment. They considered some hubs as virtual hubs to replace them once the central hub was closed. Therefore, they employed fuzzy correct linear programming [21]. Chiu and Yang worked on the designing of airlines in the Hub-and-spoke system with a random request for intermediate airports and the impact of the swarm on them. They formulated the issue with a two-step randomized programming and used China and Taiwan's airline dataset to test and assess the sensitivity of the model [22]. Harshavardhan and Krishnan proposed a framework of genetic algorithms, which use multi-level hierarchies to solve an optimization problem by searching over the space of simpler objective functions with capacities to model and solve hub locating. They used a heuristic approach based on two genetic algorithms and a hierarchical adjacency search. Their results suggested the high influence of the hierarchical genetic algorithms [23]. Using a multi-objective genetic algorithm based on game theory as well as a hybrid simulated annealing algorithm are some of the comparative optimization approaches for the NSGA-II in the bi-objective hierarchical hub location problem [24]. By proposing the combination of point-to-point and Hub-and-spoke topologies in the present paper, MOGA simultaneously finds a representative set of Pareto optimal solutions and convergences faster than its comparator algorithms for air route network optimization.

The rest of this paper is organized as follows: First, we give an overview of the point-to-point and Hub-and-spoke topologies in section II followed by the proposed research question, which aimed to optimize three goals (reducing the number of airways, the number of air stops and the flight length) simultaneously. In section III, an optimal model for airline topologies based on a multi-objective genetic algorithm has been proposed and detailed. Three object functions are introduced and the intelligent multi-object optimization model is presented. The implementation of the proposed model by genetic algorithm and the dataset specifications are described in detail in section IV, followed by the simulation results for the proposed MOGA algorithm for Iran airline topology. Then, the discussions about the research results, especially assessing the proposed model and the proposed model repeatability test are reported. At the end of the conclusions, the section states the main deductions of using a multi-objective genetic algorithm for Iran airline network optimization.

II. Point-to-Point and Hub-and-Spoke Topologies

Air route network optimization was motivated by the purpose of defining the most effective structural properties for a given air transport network, in terms of the number of airways and the aggregation of passengers in aviation industries organization, and also the changes in airways and the travel time for travelers. There are two major groups of air transportation networks:

- • Point-to-point: in this construction of the air traffic network, each justified connection pair of airports in the network is served by direct connections to all passengers with different aircraft. The number of connections and required aircrafts is about the square of the number of airports for a fully point-to-point connected network. An example of a point-to-point architecture is depicted in Fig. 1.

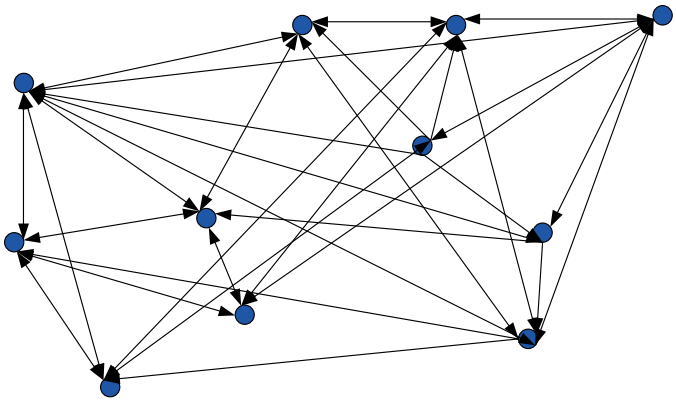


Fig. 1. The point-to-point architecture of air routes.

- Hub-and-spoke: Hub-and-spoke topology is one of the proper topologies in airway networks [25]. In this topology, instead of a direct flight between all airports, some of the terminals are considered as intermediate airports (hub airports) and other airports (spoke airports) are only connected to the intermediate airports (Hub-and-spoke architecture, Fig. 2). In this topology, instead of a direct flight between departure and destination airports, at first, all the passengers fly together from the same departures to different destinations to one of these intermediate airports. Then, they will be gone to the different departures from an intermediate airport. In the meantime, if necessary, flights between the intermediate airports will be made. In this topology with the lowest number of air routes, most airline coverage can be achieved [4].

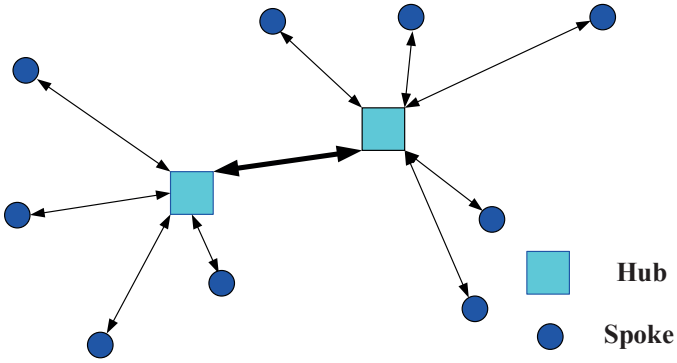


Fig. 2. The Hub-and-spoke architecture of air routes.

The point-to-point architecture of air routes, for a complete network of N airports, so that all airports can be connected (point-to-point topology), $N(N-1)$ air routes are needed. All $N(N-1)$ airlines do not have similar air traffic and on the other hand, the high number

of air routes increases the flight cost. To decrease the number of air routes, the hub-and-spoke topology can be used. Through the following equation, the total number of needed airlines in this topology with M intermediate airports from N available airports can be calculated:

$$Number\ of\ Airlines = 2(N-M) + \frac{M(M-1)}{2} \quad (1)$$

The number of reduced flights in comparison with the direct flights from all airports can be calculated from the following equation:

$$Reduction\ of\ Airlines = N(N-1) - 2(N-M) + \frac{M(M-1)}{2} = N(N-3) + \frac{M(M-5)}{2} \quad (2)$$

For instance, with 60 airports ($N = 60$) in the point-to-point topology, $60 * 59 = 3540$ airlines are needed, however, by considering an intermediate airport ($M = 1$) in the Hub-and-spoke topology, the number of airways is decreased to $2(60-1) + \frac{1(1-1)}{2} = 118$. Of course, you have to think that in the Hub-and-spoke topology, the number of air stops as well as the flight length will increase but due to the aggregation of passengers, the airplanes will carry more passengers [26]. The USA airline with two central airports located in Atlanta and Chicago or even Dubai airport as an intermediate airport in the Middle East are two examples of the Hub-and-spoke topology. This study solves the problem of optimization of all mentioned goals (reducing the number of airways, the number of air stops and the flight length) simultaneously by proposing a solution based on the combination of two topologies – point-to-point and Hub-and-spoke – with MOGA.

The optimization of topology in an airway network reduces airfare rates and boosts the flight industry. Studies have shown that in solving the problem of hub locating, first researchers concerned about parameter modeling, and then the focus has been shifted on optimal Modeling for hub problem solving and recently, the large-scale optimization issues and algorithm selection approaches have been addressed. Also, these studies acknowledged the multi-dimensional and NP-hard features of hub locating as presented in this paper.

III. The Proposed Topology of Airlines

The primary purpose of this paper is to design an optimal model for airline topologies based on a multi-objective genetic algorithm. Since the point-to-point topology and Hub-and-spoke topology have their benefits – in the point-to-point topology, the travel length is short, and the number of changes in direction is few. However, in the Hub-and-spoke topology, the number of airlines will decrease – the proposed topology of this study is defined to provide a combination of both topologies. In this topology, if necessary, an airport could connect to more than one intermediate airports or another regular airport (Fig. 3). In many studies, for the allocation of a spoke to the airport, the distance criterion has been used. In these cases, each airport has been allocated to its nearest intermediate hub.

An innovative structure based on the artificial intelligence was employed for allocating the airports to an intermediate hub by this paper. In this proposed topology, at first, the matrix of the number of passengers between airports and the matrix of the distance between airports was provided. Afterward, by using the capabilities of the MOGA algorithm, an optimal topology for the airlines was determined to achieve the supposed objectives. This model has been defined in a way to create an airline topology in which, with the slightest increase in average travel length and the average changes in paths, the number of flight routes would decrease. The outcomes of this model were presented as optimal airline topologies with the same applied value. Based on the conditions, an expert opinion, and policymakers, we could implement one of these optimal topologies. On the other hand, if due to whatever reason such as adverse weather conditions, one of these intermediate airports is excluded from this topology, we could choose another one as an intermediate airport.

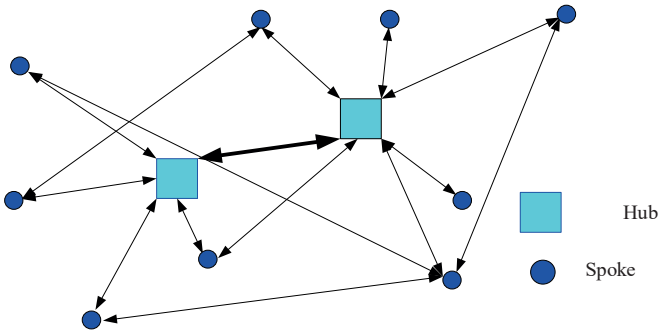


Fig. 3. The proposed combined structure of airlines (the combination of point-to-point and Hub-and-spoke architectures).

The designed model in this study comprises two main sections. One section is responsible for optimizing and finding intermediate airports, and the other section provides optimal airline topologies with the presence of the perceived intermediate airports in the first part. The flowchart of the proposed model is shown in Fig. 4.

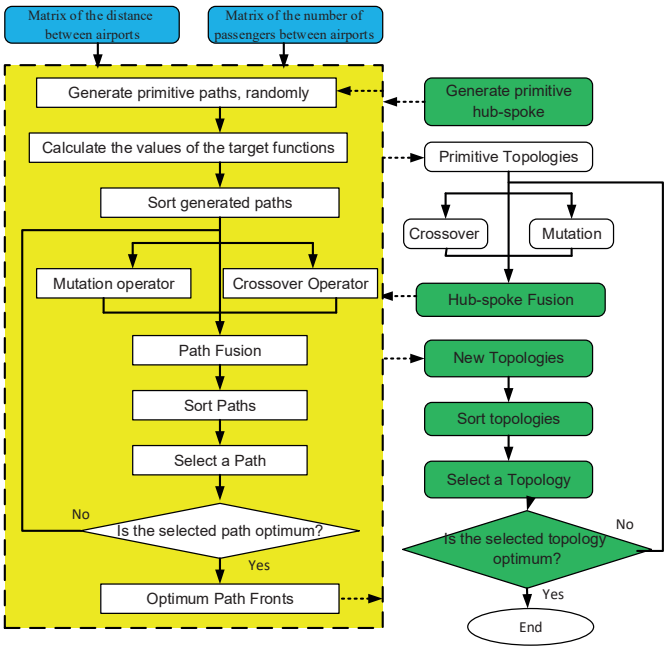


Fig. 4. The flowchart of the proposed model.

The MOGA algorithm was employed for providing optimal airline topologies in this paper. Since this model is a multi-objective one, there are multiple optimal topologies for each set of intermediate airports, which cover the outputs of optimal airline topologies. It should be noted that the amounts of each objective function for every response in the offering section of optimal topology can be calculated.

To design the airline topologies, three objective functions for simultaneously reducing the number of airlines, $Object_1$, the average travel length, $Object_2$, and the average number of air stops, $Object_3$, were taken into account that can be described as follows:

To design the airline topologies, three objective functions for simultaneously reducing the number of airlines, $Object_1$, the average travel length, $Object_2$, and the average number of air stops, $Object_3$,

were taken into account that can be described as follows:

$$\min(Object_{1,}\ Object_{2,}\ Object_{3}) \quad (3)$$

A. First Object Function: Reducing the Number of Airlines

The reduction of the number of airlines is one issue that can be considered while designing airline topologies. The highest number of airways is in the point-to-point system, and the lowest one is the Hub-and-spoke system. In this study, a combination of these two systems was employed to offer airline topologies. This target function was designed based on the following equation:

$$min(Object_1) = min\left(\sum R_{S_i \to H} + \sum R_{H \to S_j} + \frac{n_H(n_H - 1)}{2} + \sum R_{S_i \to S_j}\right) \quad (4)$$

In the above-mentioned equation, $R_{S_i \to H}$ is air route from a departure airport to an intermediate airport, $R_{H \to S_j}$ is air route from an intermediate airport to a destination airport, $R_{S_i \to S_j}$ is a direct air route between departure and destination airports, and n_H is the number of intermediate airports.

B. Second Object Function: Reducing the Average Travel Length

Another goal is reducing the average travel length, $Object_2$. This target function conflicts with the target function of reducing the number of airways, $Object_1$, because by reducing the number of airways, the journey time will increase. In other words, the shortest average travel length is in the point-to-point system, and the longest one is in the Hub-and-spoke system with one spoke. Following equation shows the implementing process of this target function:

$$min(Object_2) = min\left(\frac{\sum\sum P_{i \to j}\left(D_{S_i \to H_m} + D_{H_m \to H_n} + D_{H_n \to S_j}\right)}{\sum\sum P_{i \to j}}\right) \quad (5)$$

where $P_{i \to j}$ is the number of passengers from a departure airport (S_i) to a destination airport (S_j), $D_{S_i \to H_m}$ is the distance between the departure airport and an intermediate airport, $D_{H_m \to H_n}$ is the distance between two intermediate airports, and $D_{H_n \to S_j}$ is the distance to a destination airport from an intermediate airport.

C. Third Object Function: Reducing the Average Change in Route

To increase the welfare of the passengers and reduce the waiting time at the airport, it is necessary to reduce as many changes in routes and air stops as possible, $Object_3$. The lowest amount of changes in directions exists in the point-to-point topology. In other words, in this topology, each passenger will use only one airline from departure to destination. Following equation displays the implementation process of this target function:

$$min(Object_3) = min\left(\frac{\sum\sum P_{i \to j}\left(\delta_{S_i \to H_m} + \delta_{H_m \to H_n} + \delta_{H_n \to S_j}\right)}{\sum\sum P_{i \to j}}\right) \quad (6)$$

In the above equation, $P_{i \to j}$ is the number of passengers from a departure airport (S_i) to a destination airport (S_j). Also, H_m is an intermediate airport for departure and H_n is an intermediate airport for a destination. In this equation, if there is a flight between two airports ($S_i \to H_m$, $H_m \to H_n$ or $H_n \to S_j$), the number of δ equals to 1, otherwise, 0. It should be noted that in the two above equations, if it is supposed to be a direct flight - without an intermediate airport - from the departure airport to destination airport, S_i will be considered equal to H_m and S_i equal to H_n.

Briefly, our proposed multi-object optimization problem can be formulated as a minimization of the above-mentioned objects simultaneously which has been done by MOGA in this paper.

IV. The Implementation of the Proposed Model

In this study, to solve issues regarding the design of optimal airline topologies through the MOGA algorithm, a chromosome was first designed following the type of problem and then to perform the model, crossover, and mutation genetic operators were applied, and then appropriate choices to the kind of problem were employed. The implementation process of these matters is described as follows:

A set of chromosomes is considered as a population for the genetic algorithm [27] that evolves with its operators. In the first section of the model, a chromosome was used for encoding intermediate airports. This chromosome is encoded in binary, indicating whether an airport is an intermediate (code 1) or not (code 0) (Fig. 5).

ID Airport	1	2	3	...	n-1	n
Hubs	0	1	0	...	0	1

Fig. 5. A chromosome identifying intermediate airports among n airports.

In the second section of the model, a chromosome is considered for determining the MOGA of the connections and networks among airlines. This chromosome displays airline topologies and comprises two gene sequences along the length of the airways; the first series determines an intermediate airport for the departure airport and the second one identifies an intermediate airport for the destination airport of each airway (Fig. 6). The value of the chromosome can be obtained from the first section of the model, which included the identified intermediate airports. Some zero genes in this chromosome are taken into account for situations where an intermediate airport is not considered. If the value of a gene in two strings is zero, it will indicate that the flight between the two airports will happen directly.

ID AirLine	1	2	3	4	...	m-2	m-1	m
Hubs for Origin	7	36	0	36	...	0	7	7
Hubs for Destination	0	36	0	7	...	36	0	7

Fig. 6. A chromosome for determining of connection and network of airlines with two intermediate airports coded 7 & 36 for m airways.

One of the essential operators in MOGA algorithms is the crossover operator. It aims to explore new and especially useful areas of the research environment by replacing a part of the gene sets between the two chromosomes. In the present study, a one-point and two-point crossover operators [28] were employed. It should be noted that to create more various compounds from two parents, in the second chromosome, for each gene sequence, a crossover operator was applied separately.

The mutation operator imports new information to the population and searches in empty spaces of the problem. In this study, some genes were randomly selected to be mutated. The amount of that gene was then replaced by the values allowed for that gene.

A. Area of Study and Dataset

To test and assess the proposed model for designing Iran airline topology, the data of the Iran Airlines Industry from 2018 was taken into account. According to this data, in 2018, 18251677 passengers traveled by the air transport system with 433 airlines from 60 airports with 163407 flights. Averagely, the travel length per person was 724 kilometers, and the number of passengers per flight was 111.7. The most frequent travels were from Mehrabad Airport in Tehran. Concerning passengers' movement, Mashhad Airport was in the 2nd place. The least amount of flights belonged to Siravan and Jask Airport with only one trip. The best airport in terms of the number of travelers per flight was

Kerman Airport with 172.3 passengers per flight, and the worst airport in the country (disregarding airports with one trip) was Iranshahr Airport with only 38.2 passengers per flight. The most crowded airline was from Kerman Airport to Tehran Airport with 229.4 passengers per flight and from Tehran to Kerman with 227.6 passengers per flight. The most noticeable amount of passengers belonged to the airline from Mashhad to Tehran with 1726714 travelers, and the most significant amount of trips belonged to the same airline with 12468 flights.

The database of this study which was used for designing optimal airline topologies included statistical information on Iran airways. The statistics regarding the movement of passengers among airports in the country in 2018 got extracted from the analytical journal of the Ministry of Roads and Urban Development related to the Aviation Division and the locations of the country's airports in the Lambert coordinate system, as well as Google Earth application.

Those routes that had certain conditions such as the relocation of national authorities or relocating without passengers between two airports got removed from research. Therefore, the average passengers per airline was considered as 100. Also, the number of airlines changed from 433 to 360 efficient carriers. The Iran's passengers' movement matrix was used.

Another input of the proposed model was the matrix of air distance between airports. The most accurate method for calculating the length of the airways is Euclidean relation. Hence, at first, the spatial location of airports was calculated in Lambert coordinate system, and then the following equation was used for calculating air route length.

$$Distance(A,B) = ((X_A - X_B)^2 + (Y_A - Y_B)^2)^{\frac{1}{2}} \quad (7)$$

In the above equation, X_A and Y_A are the geographical length and width of Airport A (departure airport), and X_B and Y_B are the topographical length and breadth of Airport B (destination) in Lambert coordinate system. After obtaining the distance between airports, the matrix of airport distances was provided.

V. Simulation Results

The output of the proposed multi-objective genetic algorithm offered multiple optimal airline topologies (a combination of point-to-point system and Hub-and-spoke system) that are equal from the optimization value and the final optimal architectures can be selected and applied based on the policies in the field of air industry and experts' opinions.

For the described dataset, one of the model outputs regarding optimal airways which includes Mehrabad and Mashhad airports as intermediate airports, is the growth of the average travel length up to 13.8% (about 100 kilometers) and also by increasing the average changes in routes up to 18%, 50.8% of air routes (184 air routes) will decrease compared to the current situations. Moreover, in another case with the same air intermediate, by increasing the travel length to 4.5% (32 kilometers) and the number of changes in routes to 5%, 24.7% of the air routes (89 airways) will decrease. Table I shows some of the topologies in model output.

The number of airways, and the average travel length for the optimal topology with two target functions is shown in Fig. 8-A. Fig. 8-B shows the number of airways and changes in routes for the proposed algorithm with two target functions. There is a reverse relationship between the target function of numbers of airways and other target functions – average travel length and the average number of changes in routes. On the other hand, this optimal topology indicates there is a direct relationship between the target function of average travel length and average changes in directions. Fig. 7 shows some of the topologies of model output.

The optimal answers to this problem have a three-dimensional space in the number of target functions. To have a better display of the answer curve, optimal solutions for target functions were offered two-to-two in a 2D environment (Fig. 8-A, B & C) and the rest of the target functions were presented in a 3D environment (Fig.8-D).

According to the output results, the proposed model can offer optimal answers and is significantly better than the current situations of the country's airways. The presence of different answers with intermediate airports indicates the model's capability in providing a combined airway topology from the combination of point-to-point and Hub-and-spoke topologies (e.g., answers No. 4 & 5 in Table I). For allocating air routes between an airport and one of the intermediate airports in the proposed model, Genetic algorithm optimization methods were employed instead of considering the nearest intermediate airport. Of course, in many cases, the nearest intermediate airport to each airport has been selected as the intermediate airport in that airport, however, not always. For example, if Mashhad airport and Isfahan airport are chosen as an intermediate airport, based on the distance, it is expected that Mashhad airport would be the intermediate airport for Iranshahr airport. However, given the results of model output, it can be seen that Isfahan airport with a more significant distance has been selected as an intermediate airport for Iranshahr airport. In the rest of this section, the model structure will be assessed.

A. Assessing the Proposed Model

In this study, the convergence plot and the path non-constrained-

Table I. Number of Optimal Airline Topologies of Model Output

No.	Aviation centers	Number of airways	Average travel length	Average air stop	Number of direct flights
1	__	360	7/747	1	362
2	Mehrabad	115	897	1.26	0
3	Mehrabad – Isfahan	115	870	1.74	0
4	Mehrabad – Mashhad	139	882	1.21	2
5	Mehrabad – Mashhad	178	851	1.18	32
6	Mehrabad – Mashhad	271	780	1.05	113
7	Mehrabad – Mashhad – Shiraz - Ahvaz	250	777	1.13	45
8	Mehrabad – Mashhad – Isfahan – Shiraz – Ahvaz	131	817	1.42	0
9	Mehrabad – Mashhad – Dezfool – Shiraz – Ahvaz – Isfahan – Bandarabas	142	833	1.4	0
10	Mehrabad – Mashhad – Dezfool – Shiraz – Ahvaz – Isfahan – Yazd	209	770	1.18	4

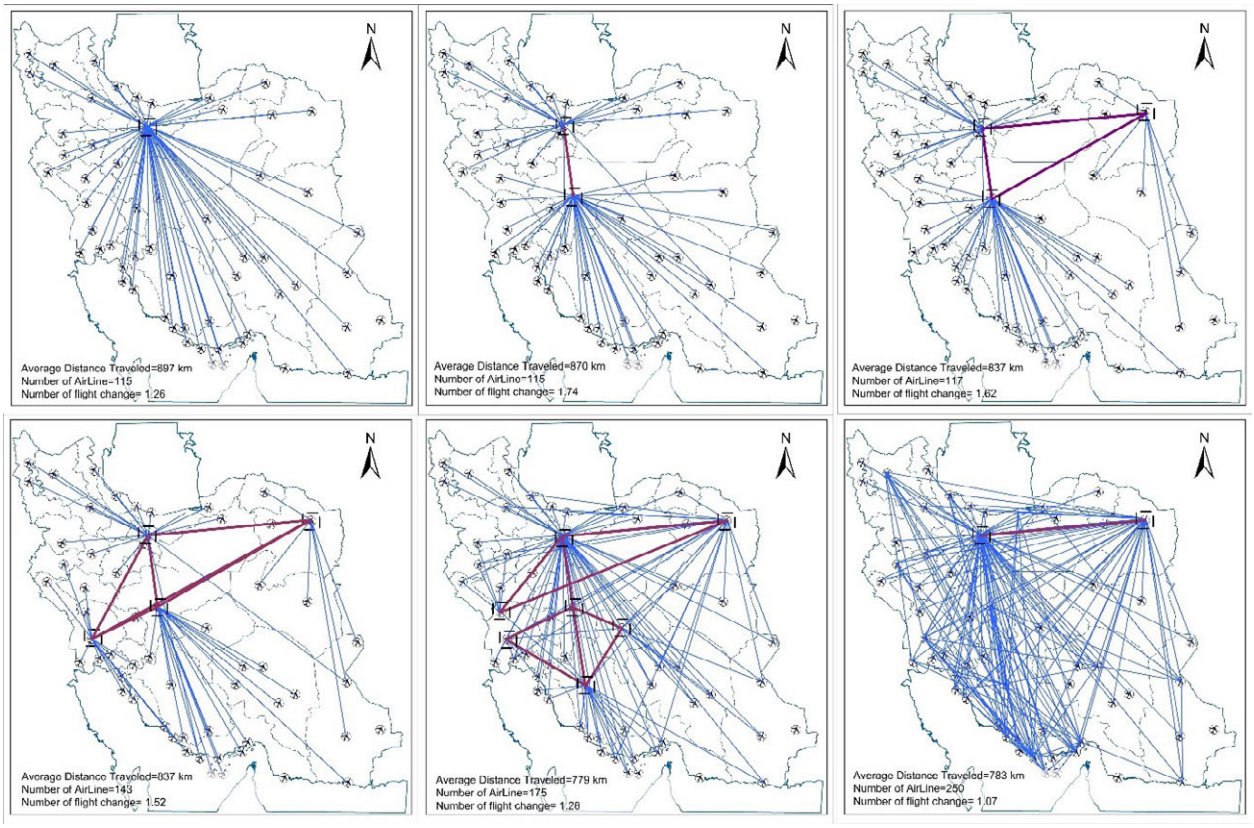


Fig. 7. Some of the airline architectures created by the proposed multi-object optimal topology.

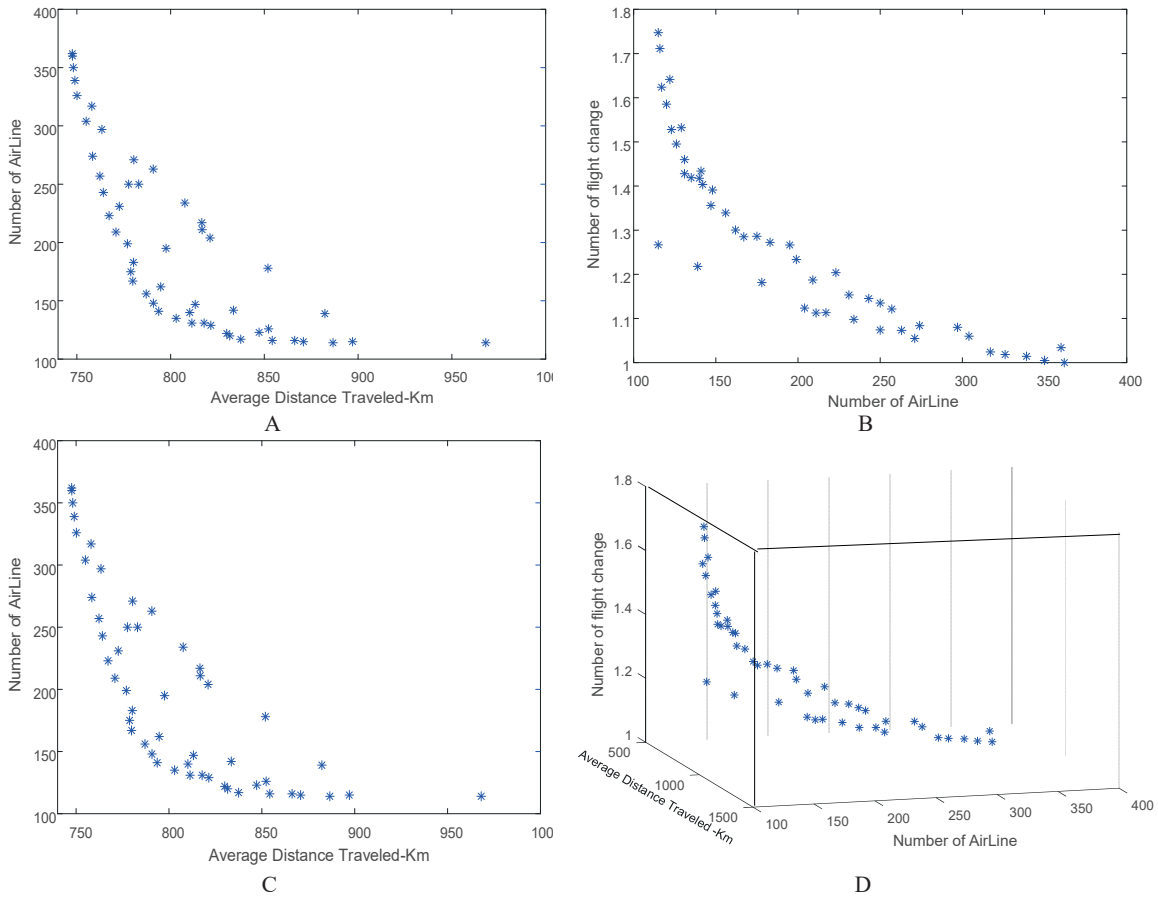


Fig. 8. The optimal answers to the optimization airway network problem by the proposed model in three-dimensional space based on the value of target functions (Number of Airline vs. Number of flight change vs. Average travel length in Km).

domination chart (the model of variation of the number of airline vs. number of flight change vs. average travel length in Km) were used for assessing the proposed model. Convergence calculates the area covered by the answers to the parade (located in the target space) and path charts demonstrate the values of the optimal topology response set [29] in each of the target functions. The variations of the convergence plot of the proposed model for the supposed target functions have been normalized and can be seen in Fig. 9. This graph shows the qualitative evaluation of the obtained responses regarding each objective function.

As it is obvious in Fig. 9, the designed optimal model in this study has been able to improve the topology during different generations and move to optimal results. This matter suggests the possibility of discovering optimal answers in the designed model. Since the 300th generation so far, the ascending slope of the growing trend of the magnitude of the dispersion criterion has decreased, which indicates the convergence of the outputs of the proposed model to optimal solutions. The value chart of the solutions of the 500th generation achieved from the model is shown in Fig. 10. This figure was drawn for the normalized target functions.

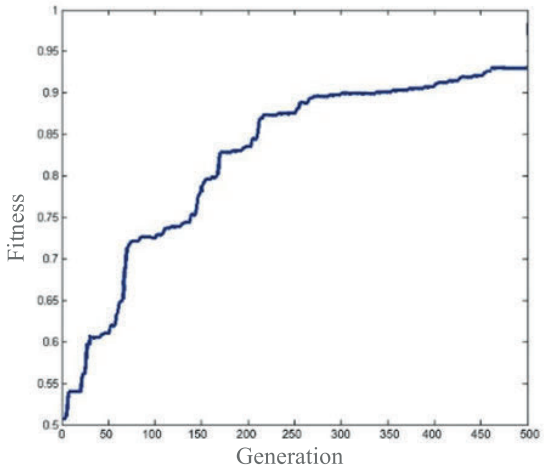


Fig. 9. The convergence plot of the proposed model

As it is shown in Fig. 10, the results obtained from the proposed model are expanded in each of the object functions throughout its target length, which indicates the acceptable dispersion of the results. Major changes are also observed in the slope of the routes between the two object functions, the number of airways and the average travel length, that suggest a good exchange between unsettled answers. On the other hand, significant changes in the gradient of the response slope are not observed in both functions – average travel length (target function # 2) and average changes in routes (target function # 3) – which indicates the presence of a correlation between these two target functions.

B. Model Repeatability Test

Since in multi-objective evolutionary algorithms, the search start point is chosen randomly, the results will vary with different performances. To test its repeatability, the model was performed four times with a population of 50 individuals with 300 repetitions with similar parameters. Fig. 11 shows the results from four times of repetition for the target function of the number of airways (Fig. 11-A) and average travel length (Fig. 11-B). As shown in the diagrams of this figure, the model has had a steady trend in all four repetitions, and the model outputs are almost located in a given range of target functions. The results are shown in Fig. 11 that compares the number of airways and average travel length, as two of the target functions, to present repeatability of the algorithm in the global optimization process. In the results of the four runs, there are similarities and differences between the number of airways and average travel length of the global solutions found based on the different selections of 50 individuals with 300 repetitions. From the investigation of the results, we found that the variance for the number of airways is in the range of 10%, and it ranged between 5% for average travel length. From these results, it was concluded that the average repeatability of the proposed algorithm for different selections of individuals is about 93%.

VI. Conclusions

By studying Iran airlines, we understand that they can get more optimal. The proposed model outputs suggest some intermediate airports based on the Iran's aviation transport matrix. Therefore, there will be fewer airlines, and we can establish connections between airports with lesser money. Given the results of this paper, by considering a combination of point-to-point and Hub-and-spoke topologies, we would be able to offer more efficient airways. The designed model of this article tries to connect one airport to some intermediate airports,

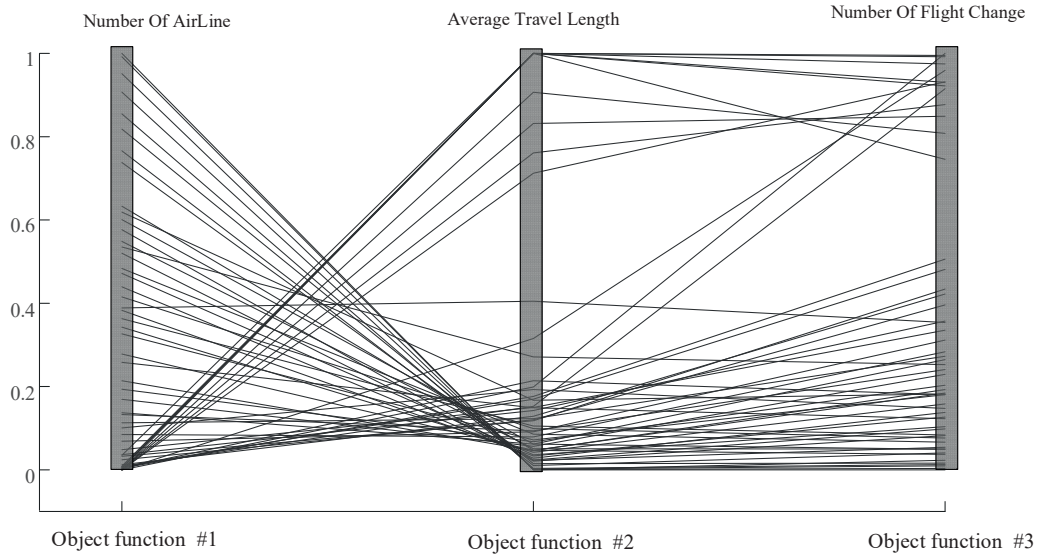


Fig. 10. The non-constrained-domination model (the model of variation of number of airways vs. number of flight change vs. average travel length in Km) of the 500th generation.

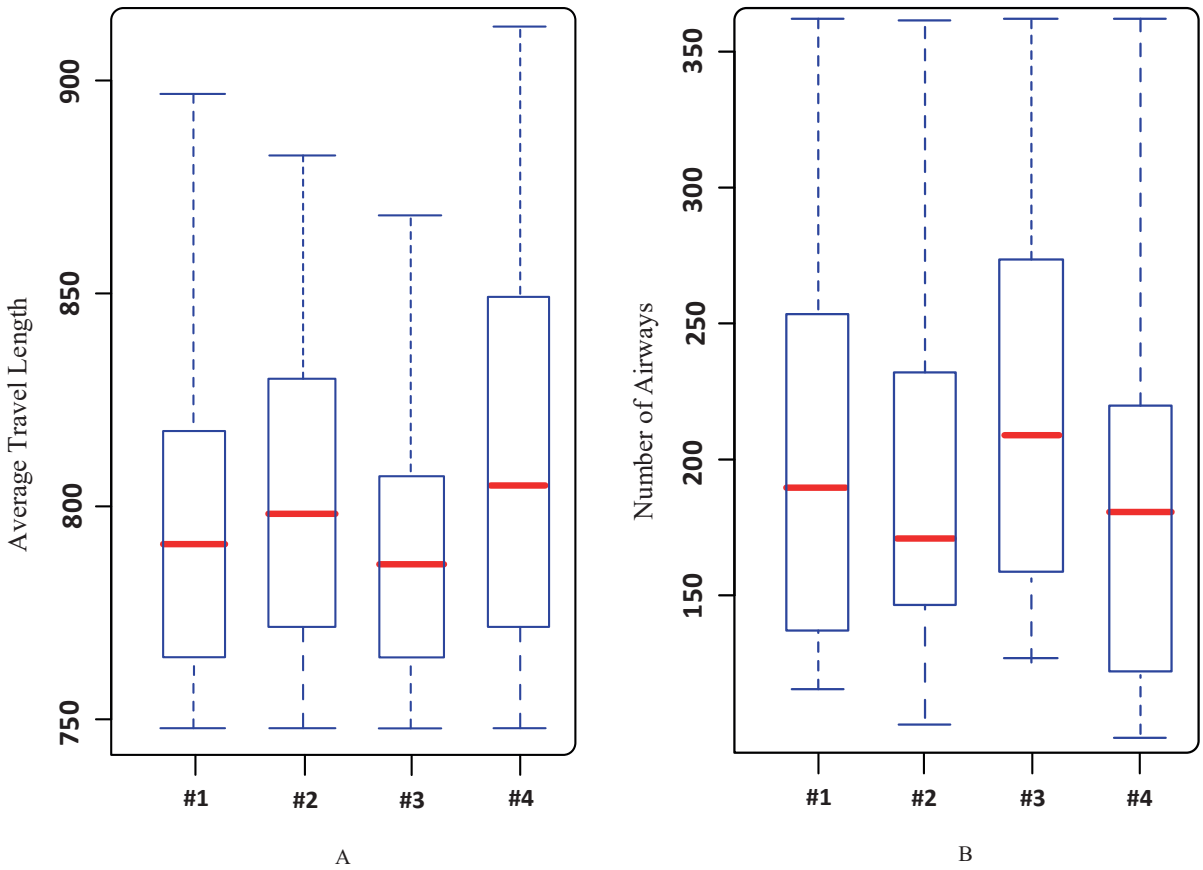


Fig. 11. Variations in the two target functions: average travel length (A), and the number of airways (B) and in different repetitions.

which are an excellent method to have more efficient airline topologies in the country. It should be noted that in airline topologies with the presence of one or more intermediate airports, besides having less populated airplanes and less expensive flights, we will have some economic benefits such as the ability to build convenience stores and economic prosperity in intermediate airports (such as Dubai Airport). We should remember that although airline topologies with intermediate airports increase changes in routes and therefore the travel length, with the aggregation of airways, the number of flights in an airway will increase and passengers could choose their flight hours during the day. One of the benefits of the designed model in this study is its capability in offering several options for airline topologies, and it lets managers and decision-makers select and apply one of the optimal topologies based on the current conditions. This feature also allows managers to replace another optimal topology if one or some intermediate airports are down because of whatever reason such as adverse weather conditions or something else. Scheduling flights in aviation topology design should be considered in future studies. So that the waiting time for a passenger in intermediate airports is viewed as a target function and efforts are focused on reducing it. Moreover, we should pay more attention to the capacity of air transport systems and airports and think of it as a limitation. Weather conditions as well as atmospheric environment matter while choosing an intermediate airport. It should be noted that, for some reason, such as policy domination or economic problems, airports that are not optimal will be viewed as intermediate airports. For instance, Dubai Airport is a convenient airport for the Middle East. Hence, it is crucial to consider the political and economic conditions in the selection of intermediate airports, which is recommended in future studies of this case too.

References

[1] A. Fraile, J. Sicilia, R. González, and A. González, "Study of Factors Affecting the Choice Modal of Transportation in an Urban Environment Using Analytic Hierarchy Process," in *Applied Computer Sciences in Engineering*, Springer International Publishing, 2017, pp. 357-367.

[2] M. E. O'Kelly, "A quadratic integer program for the location of interacting hub facilities," *European Journal of Operational Research*, vol. 32, no. 3, pp. 393-404, 1987.

[3] P. Bagga, A. Joshi, and R. Hans, "QoS based Web Service Selection and Multi-Criteria Decision Making Methods," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 4, pp. 113-121, 2019.

[4] L. Mei, Z. Yan and J. Shao, "p-hub median location optimization of hub-and-spoke air transport networks in express enterprise," *Special issue on multimodal information learning and analytics on cross-media big data, Concurrency and Computation: Practice and Experience*, *Wiley Online Library*, vol. 31, no. 9, 2019.

[5] O. Lordan, J. M. Sallan and P. Simo, "Study of the topology and robustness of airline route networks from the complex network approach: a survey and research agenda," *Journal of Transport Geography*, vol. 37, pp. 112-120, 2014.

[6] C. Liu and Q. Gui, "Mapping intellectual structures and dynamics of transport geography research: a scientometric overview from 1982 to 2014," *Scientometrics*, vol. 109, no. 1, p. 159–184, 2016.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[8] S. Garcia-Rodriguez, "Application of Multiobjective Evolutionary Techniques for Robust Portfolio Optimization," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 2, no. 2, pp. 63-64, 2013.

[9] M. Borhani and N. Ghasemloo, "Soft Computing Modelling of Urban Evolution: Tehran Metropolis," *International Journal of Interactive*

Multimedia and Artificial Intelligence, p. In Press, 2020.

[10] A. H. A. Elkasem, S. Kamel, A. Rashad, and F. Jurado, "Optimal Performance of Doubly Fed Induction Generator Wind Farm Using Multi-Objective Genetic Algorithm," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 5, pp. 48-53, 2019.

[11] A. Taibi and B. Atmani, "Combining Fuzzy AHP with GIS and Decision Rules for Industrial Site Selection," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, no. 6, pp. 60-69, 2017.

[12] M. Borhani, "Corpus Analysis Using Relaxed Conjugate Gradient Neural Network Training Algorithm," *Neural Processing Letters*, vol. 50, no. 1, p. 839–849, 2019.

[13] M. Borhani and H. Ghassemian, "Hyperspectral spatial-spectral feature classification based on adequate adaptive segmentation," in 2014 *Iranian Conference on Intelligent Systems (ICIS)*, 2014.

[14] J. Sicilia, C. Quemada, B. Royo, and D. Escuín, "An optimization algorithm for solving the rich vehicle routing problem based on Variable Neighborhood Search and Tabu Search metaheuristics," *Journal of Computational and Applied Mathematics*, vol. 291, pp. 468-477, 2016.

[15] A. Fraile, E. Larrodé, A. Magreñan, and J. Sicilia, "Decision model for siting transport and logistic facilities in urban environments," *Journal of Computational and Applied Mathematics*, vol. 291, no. C, pp. 478-487, 2016.

[16] J. Sohn and S. Park, "A linear program for the two-hub location problem," *European Journal of Operational Research*, vol. 100, no. 3, pp. 617-622, 1997.

[17] V. Marianov and D. Serra, "Location models for airline hubs behaving as M/D/c queues," *Computers & Operations Research*, vol. 30, no. 7, 983–1003.

[18] B. Wagner, "An exact solution procedure for a cluster hub location problem," *European Journal of Operational Research*, vol. 178, no. 2, p. 391–401, 2007.

[19] M. Mohammadi, F. Jolai and H. Rostami, "An M/M/c queue model for hub covering location problem," *Mathematical and Computer Modelling*, vol. 54, no. 11, pp. 2623-2638, 2011.

[20] S. A. Alumur, H. Yaman and B. Y. Kara, "Hierarchical multimodal hub location problem with time-definite deliveries," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 6, pp. 1107-1120, 2012.

[21] F. Taghipourian, I. Mahdavi, N. Mahdavi-Amiri and A. Makui, "A fuzzy programming approach for dynamic virtual hub location problem," *Applied Mathematical Modelling*, vol. 36, no. 7, pp. 3257-3270, 2012.

[22] T. H. Yang and T. Y. Chiu, "Airline hub-and-spoke system design under stochastic demand and hub congestion," *Journal of Industrial and Production Engineering*, vol. 33, no. 2, pp. 69-76, 2016.

[23] P. K. Harshavardhan and K. Kousik, "Hierarchical Genetic Algorithms with evolving objective functions," *Neural and Evolutionary Computing*, p. arXiv:1812.10308, 2018.

[24] M. Khodemani-Yazdi, R. Tavakkoli-Moghaddam, M. Bashiri, and Y. Rahimi, "Solving a new bi-objective hierarchical hub location problem with an M/M/c queuing framework," *Engineering Applications of Artificial Intelligence*, vol. 78, pp. 53-70, 2019.

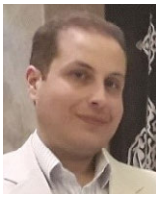
[25] D. L. Bryan. and M. E. O'Kelly, "Hub-and-Spoke Networks in Air Transportation: an Analytical Review," *Journal of Regional Science*, vol. 39, no. 2, pp. 275-295, 1999.

[26] S. W. C. Morrison, The economic effects of airline deregulation, Washington, DC.:Brookings Institution Press., 2010.

[27] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

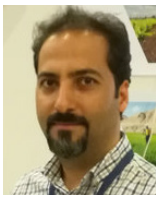
[28] R. L. Haupt and S. E. Haupt, Practical genetic algorithms, John Wiley & Sons, 2004.

[29] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, New York, NY, USA: John Wiley & Sons, Inc., 2001.



Mostafa Borhani

Dr. Mostafa Borhani is an expert at Communication System design, received the BSEE degree and MSEE degree from Sharif University of Technology, in 2002 and 2004, respectively and Ph.D. at the Department of Electrical and Computer Engineering, Tarbiat Modares University in 2015. He began his collaboration with high-tech PPPs and cooperated as a public-sector authority and advisors in some private-sector companies, time by time. He managed some national megaprojects such as DWDM national optical transport network (Tadbir) and IRIB Shima project. He is responsible as General Director of Administering and Leadership of OTT Service Operators in Cyberspace. His professional background includes scientific, research, and executive activities in advanced of structured networking, international relations, and the commercialization of knowledge based products. Artificial intelligence, computational intelligence, and data mining are some of his recent activities concerning extensive machine learning applications in sports, cyberspace and Holy Quranic data analysis to achieve more comprehension and understanding on this holy book. His research is conducted as faculty member of the Quran Miracle Research Institute of the Shahid Beheshti University.



Kamal Akbari

Kamal Akbari is currently a Ph.D. Candidate in Remote Sensing and GIS center of Shahid Behehshti University, as well as a spatial data analyst in GIS at GITA. He has over ten years of wide experience in GIS, Spatial Database Designing, Pedestrian Modelling, and Teaching. Kamal is an expert in ArcGIS, QGIS, FME ETL tool, R programming and Python.

Aliakbar Matkan

Aliakbar Matkan is a Professor of Earth Sciences and Head of Remote Sensing and GIS Research Center, Faculty of Earth Sciences, Shahid Beheshti University.

Mohammad Tanasan

Mohammad Tanasan is currently a Ph.D. Candidate in Remote Sensing and GIS center of Shahid Behehshti University.

Finding an Accurate Early Forecasting Model from Small Dataset: A Case of 2019-nCoV Novel Coronavirus Outbreak

Simon James Fong[1,2]*, Gloria Li[2], Nilanjan Dey[3]*, Rubén González Crespo[4], Enrique Herrera-Viedma[5]

[1] Department of Computer and Information Science, University of Macau, Macau SAR (China)
[2] DACC Laboratory, Zhuhai Institutes of Advanced Technology of the Chinese Academy of Sciences (China)
[3] Department of Information Technology, Techno India College of Technology (India)
[4] Universidad Internacional de La Rioja, Logroño (Spain)
[5] University of Granada (Spain)

Received 5 February 2020 | Accepted 7 February 2020 | Published 7 February 2020

Abstract

Epidemic is a rapid and wide spread of infectious disease threatening many lives and economy damages. It is important to fore-tell the epidemic lifetime so to decide on timely and remedic actions. These measures include closing borders, schools, suspending community services and commuters. Resuming such curfews depends on the momentum of the outbreak and its rate of decay. Being able to accurately forecast the fate of an epidemic is an extremely important but difficult task. Due to limited knowledge of the novel disease, the high uncertainty involved and the complex societal-political factors that influence the widespread of the new virus, any forecast is anything but reliable. Another factor is the insufficient amount of available data. Data samples are often scarce when an epidemic just started. With only few training samples on hand, finding a forecasting model which offers forecast at the best efforts is a big challenge in machine learning. In the past, three popular methods have been proposed, they include 1) augmenting the existing little data, 2) using a panel selection to pick the best forecasting model from several models, and 3) fine-tuning the parameters of an individual forecasting model for the highest possible accuracy. In this paper, a methodology that embraces these three virtues of data mining from a small dataset is proposed. An experiment that is based on the recent coronavirus outbreak originated from Wuhan is conducted by applying this methodology. It is shown that an optimized forecasting model that is constructed from a new algorithm, namely polynomial neural network with corrective feedback (PNN+cf) is able to make a forecast that has relatively the lowest prediction error. The results showcase that the newly proposed methodology and PNN+cf are useful in generating acceptable forecast upon the critical time of disease outbreak when the samples are far from abundant.

Keywords

Forecasting, Machine Learning, Method, Prediction, Data Mining, Epidemic.

DOI: 10.9781/ijimai.2020.02.002

I. Introduction

SINCE December 2019, the first case of human infection by a novel (unknown and new) virus which was formerly and now known as Wuhan virus and Coronavirus coded as 2019-nCov respectively, was reported [1]. As it was speculated that 2019-nCov originated from a single wild-animal which was traded at a busy marketplace [2], the toll of infested people sharply arose in the city Wuhan, spreading to other Chinese cities and eventually becoming a global epidemic within merely a month. 2019-nCoV is found to be highly contagious via bodily droplets in air, the virus can live up to hours to two days when left over a contacted surface. Human-to-human transition has recently been detected, and the toll of the infested is rising up almost exponentially in this early stage. A global health emergency warning was issued by the World Health Organization on 30 January 2020 [3], designating that 2019-nCoV is of an urgent global concern. At the early stage, the morbidity mortality rates resulted from the infection of 2019-nCoV are uncertain [4] especially for young children and the senior aged groups. In order to control the wide and fast spread of the virus, authorities took pre-emptive measures to cordon off infested cities. Such measures include closing borders, suspending community services and schools, minimizing both domestic and international travels etc., until further notice. The purpose is to limit the chances of physical contacts among people, so to thwart the transmission of the novel virus.

The curfew imposed for China and other countries will cost huge economical loss increasingly every day. As this is a novel virus, its severity is unforeseen albeit its contagiousness is very strong and the incubation period is relatively longer than other virus, it is difficult for anybody to decide the optimal time of lifting the ban. Too soon should the ban be lifted; the epidemic may not have totally subsided; prolonged extension in constraining the societies will lead to heavier

* Corresponding author.
E-mail addresses: ccfong@um.edu.mo (S. J. Fong), nilanjan.dey@tict.edu.in (N. Dey).

economical loss. Timing is very uncertain during this initial stage provided that the virus is novel, and we all have little knowledge about its characteristics. Authorities want to know when this epidemic may end, and whether it is continuing to get worse.

Therefore, forecasting is extremely important even for the slightest clues for multi-attributes consideration over other societal and public health factors. In this case, just a forecast from any single model is not enough, the most reliable forecast is needed often from selection of several candidate models. In data mining this is a challenging computational problem [5] [6] - how to build a most accurate forecasting model with only a handful of training data available in the early stage? Three popular approaches of building prediction model from a small dataset are reported in the literature [7] [8]. One way is to expand the training dataset by augmenting additional data onto the available data [9] [10]. The second way [11] is to use an ensemble for collectively generating forecast results, and the forecast result from one of the several algorithms that has the lowest error is selected to use. The results from the rest of the candidates are discarded. The third approach [12] is to stay focus on a single prediction algorithm which often comes with several parameters to be adjusted. The accuracy level of the resultant model tends to be very sensitive to parameter setting. The default parameters values for such algorithm often do not provide the maximum performance, fine-tuning at the parameters values is required to improve the accuracy level.

In view of the gravity that associated with decisions to be made about the epidemic, a new data mining methodology should be considered as an alternative to the existing ones on inferring the best model from few training data. In this paper, the newly proposed methodology embraces the merits of the three methods with the following features: 1. multiple candidate forecasting algorithms are put together for group prediction, the one that has the lowest error is selected; 2. each of the forecasting model is tuned with the most suitable parameter values; and 3. relevant information about the prediction target are added in multiple regression type of forecasting models, as one of the group selection candidates.

The reminder of the paper is structured as follow. Section two describes the new group forecasting model with early stage small dataset samples. The experiment is presented in Section three, with discussion of results follows on Section four. Section five concludes the paper as well as sharing with future works for follow-up research on this critical topic.

II. Our Proposed Methodology

In the early stage of decision making for fast developing epidemic, little data is available, our forecasting model needs to deal with high uncertainty. It is assumed that the virus is novel and human expert judgement will be consulted after a technical forecast is made, similar to Delphi decision making. A methodology for data mining using small dataset is needed in consideration of three main objectives - the adopted forecasting model must be most competitive compared to its peers (with the lowest error), the wining model itself is optimized to its maximum performance, and the wining model has the flexibility of including other relevant time-series for multiple regression.

Our proposed methodology is designed with respect to offering the highest possible level of prediction accuracy under the constraints of low data availability and knowledge. The design focus on doing group forecasting with a collection of optimized forecasting models, some of which are able to take multiple data sources as inputs. The methodology is therefore named as Group of Optimized and Multi-source Selection, abbreviated as GROOMS.

A. Group of Optimized and Multi-source Selection

As a methodology, GROOMS has three main processes. Firstly, the full dataset, however small it may be, is passed to a collection of candidate forecasting models. Secondly each candidate model will be tuned to its best performance by optimizing its parameters values or inclusion of data sources. In some special case of multiple regression, more than one data sources which are related to the prediction target are fed into the inputs of a multi-variable prediction model for reasoning. Neural network is one typical example. Depending on the characteristics of the forecasting model, which are roughly classified into three: non-parametric (e.g. linear regression, simple statistical inference) that requires no setting of model parameter; parametric model (e.g. choice of activation function, learning rate, for neural network, entropy, pruning option and splitting criteria for decision tree) in which the output performance is sensitive to the choice of the parameters values; and dual models whose performance is very sensitive to both choices of input sources and the model parameters setting. The steps and processes of GROOMS are shown in Fig. 1.

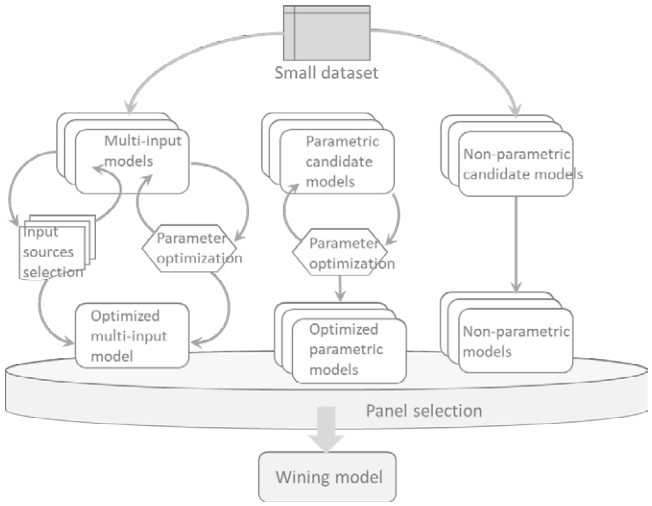


Fig. 1. GROOMS with top-down optimization processes.

In GROOMS, the available small dataset is passed from top to down through optimization processes, preparing the candidate models for panel section. Taking the curve fitting error by a forecasting model as the prime performance indicator, the panel selection picks a winning model that has the lowest error. The prediction result from the winning model is hence considered as the best effort prediction out of what technically is available using simple desktop forecasting technology [13].

Fig. 2. GROOMS in recommended working order and complexity.

In general, there are five groups of data analytics pertaining to constructing a forecasting model. As shown in Fig. 2, five groups embrace different types of algorithms and they have different complexity in the model designs [14].

Group 01: Forecasting using complex machine learning models with multiple regressions.

This is usually the neural network type of models which have highly configurable structures and parameters, they usually work on multi-variate input data. When the number of attributes or sources of the input data is high, the data is said to have high dimensionality and feature selection [15] may need to be applied. In the context of early stage epidemic forecasting where high dimensional data may not be so available, feature selection is a matter of manually selecting a combination of appropriate time-series to feed into the learning model. This group of candidate models requires dual tuning of model parameters together with the selection of a subset of time-series for minimizing the prediction errors.

Group 02: Forecasting using complex machine learning models.

This is referring to forecasting models which use relatively complex machine learning models as base learners. The input data are ordered with timestamps, the base learners attempt to learn the patterns and output predictions or forecasts by regression. Some example is support vector machine or neural network which requires iterative hyperplanes/weights updates for maximizing the results. Normally this type of models consumes considerable amount of memory and runtime.

Group 03: Forecasting using light-weight machine learning models

Similar to the one about, but it uses simple machine learning models as base learners. They run faster and generate results almost as quickly as traditional time-series forecasting. However, underfitting and overfitting can easily occur. Parameter tuning should be done to assure a reliable model.

Group 04: Simple data analytics

Mostly are descriptive statistics, e.g. average, min, max, the increments since yesterday or some days ago, rate of increases of suspected cases in comparison to cured cases, etc.

Group 05: Econometrics type of time-series forecasting

Traditional econometrics such as auto regression, exponential regression, moving average, ARIMA and its variants etc. These methods usually work over univariate time-series with none or little parameter. For example, window size for moving average, and ratios between autoregression and moving average for ARIMA, etc.

The five groups of methods have various complexity therefore run-time and modeling time which may require manually fine-tuning which takes time. In view of urgency when a quick but most reliable possible forecast is required, it is suggested that in GROOMS the groups of methods should be attempted in priorities in this sequence, as shown in Fig. 2. That is, traditional forecasting from group 5 should be first applied, probably at the same time with group 4 statistical analysis. When resources and spare time are available, methods from groups 3, 2 and 1 could be attempted in order for pursuing a lowest cost forecasting model under time constraints.

B. Polynomial Neural Network with Corrective Feedback (PNN+cf)

For group 1 - Forecasting using complex machine learning models with multiple regressions, wide choices of candidate models are available. They range from a simplest single perceptron learner to sophistical deep learning models that have many configuration options. In the literature there are state-of-the-arts that discussed using heuristic techniques for hyper-parameter optimization. It is generally known that the more parameters involved in a machine learning model, the more variables to the influences of the ultimate performance. Epidemic need quick decisions, and it could be a tedious task to try out every combination of parameter values for every model even using semi-automated approach. To this end, an alternative neural learning approach called Polynomial Neural Network is chosen in lieu of mainstream convolution neural network.

For this forecasting task of early stage epidemic, an evolving deep learning model is used. In an early stage, we neither know about in advance how long nor how fluctuating the time-series will be as time goes by. Likewise, it is not known and therefore should not permanently fix or limit about the structure of a neural network in terms of how many hidden layers and number of neurons in each layer that the network should scale up to. Therefore, an evolutionary type of neural network is desirable for such situation.

In this study, a type of evolutionary neural network called polynomial neural network (PNN) which is based on the classic principle of Group Method Data Handling (GMDH) is used. PNN expands the neural network complexity iteratively until no more performance improvement by further addition of neuron to the network structure is observed.

The original concept which was invented in 1968 by Ivakhnenko [16], is known as one of the oldest prototypes of convolution neural network. PNN is mainly powered by idea of complicating a polynomial equation which grows a neural network with scaling up the powers of the polynomial coefficients until it can model the time series as fit as possible. As a result, the neural network would remain at marginal model complexity while its predictive ability is at its highest. PNN grows the network structure by self-tuning the model's coefficients (parameters) automatically through an iterative data sampling and controlled network expansion process. The network size is therefore increasing one iteration at a time. This is a typical non-linear optimization process between the inputs and the output of the network. In its simplest form, the optimization argument by increasingly complicating the current model is as follows:

$$\bar{\beta} = \arg\min_{\beta \subseteq B} Output\_error(\beta) \tag{1}$$

where $Output\_error(\beta)$ is a criterion of output error of the model β, which will be iteratively tried from a full space of candidate beta models B. The candidate beta models are the components that build up a final PNN which piece themselves into a nonlinear multi-parametric equation. The equation $\beta=f(x_1, \ldots, x_n)$ maps the non-linear relation among a range of sequential variables, $x_{t=1}$,., $x_{t=n-1}$, x_n which is comprised as an input vector $\bar{x}$, and the predicted outcome is y from the final model $\bar{\beta}$.

Hence a PNN is simply a polynomial equation whose coefficients and their powers are modelled as weights and neurons. To compute the values of the coefficients for solving the polynomial equation, a polynomial reference function which is Equation 2 is used that represents the different candidate models that grow in PNN optimization process as follows:

$$y = \varepsilon_0 + \sum_{t=1} \varepsilon_{t=1} x_{t=1} + \sum_{t=1} \sum_{t=2} \varepsilon_{t=1,t=2} x_{t=1} x_{t=2} + \sum_{t=1} \sum_{t=2} \sum_{t=3} \varepsilon_{t=1,t=2,t=3} x_{t=1} x_{t=2} x_{t=3} + \cdots \tag{2}$$

Equation 2 shows a classical Kolmogorov-Gabor polynomial which resembles a functional series for PNN; it is able to represent any function in a general form $y=f(\bar{x})$ with its coefficient vector $\bar{\varepsilon}$. $\bar{\varepsilon}$ is the solution which PNN tries to obtain by solving the regression polynomial. As time goes that is modelled by iteration, the values of variables $\bar{x}$ feed to PNN, it computes the coefficients at the same time trying to incrementally increase the complexity. This is controlled by monitoring the error level. When there is no significant incremental gain in performance, y is then taken as the forecast result.

There are two optional forms of PNN, one being Combinatorial PNN (Combi) [17] and the other one is Multilayer Iterative PNN-type Neural Network (MIA). Combi simulates the polynomial formulae

by a simple combinatorial function that rolls up the power iteratively and extends the polynomial equation with higher power. MIA [18] is modified from a standard feed-forward neural network which is constructed initially from bi-input neurons. The neurons and layers expand by some neuron selection criteria in a way similar to the design reported in [19].

PNN has been applied for predictions of massive earthquake disasters [20], energy consumption [21] and gas furnace behaviours [22] which all are quite erratic and hard to predict in nature. To further improve the capability of PNN, a new version called PNN with corrective feedback (PNN+cf) is proposed.

PNN+cf allows feeding in extra input variables during the formation of polynomials. However, it should be carried out carefully because redundant variables can lead to overfitting, thereby degrading accuracy, and incurring long computational time cost. Only statistically significant or so-called relevant extra information should be added for accuracy enhancement. In this study two extra information, lagged data and training errors from past iterations of model training which are known as corrective feedbacks are added. The feedbacks are 1-step lag which appears like a shadow of the time series and residuals. The motivation for adopting residuals which are the deviations between the predicted and actual values, is the compensation of bias towards the nonlinearity that exists in the time series.

An illustration of the fusion of the two extra inputs to a neural network is shown in Fig. 3. The inputs from the original time series, values from the lagged data series, and residual information are augmented together. A rolling window of certain size temporarily feeds a part of the training prior to the neural network, and it predicts an outcome one step ahead. At each prediction, the residuals from previous training cycles are fed back to the neural network. By this way, the regression process embraces both the fresh input data while adjusting their weights that are biased away from the residuals. A better fitting curve is resulted from the polynomial equation for regressing a low-error output.

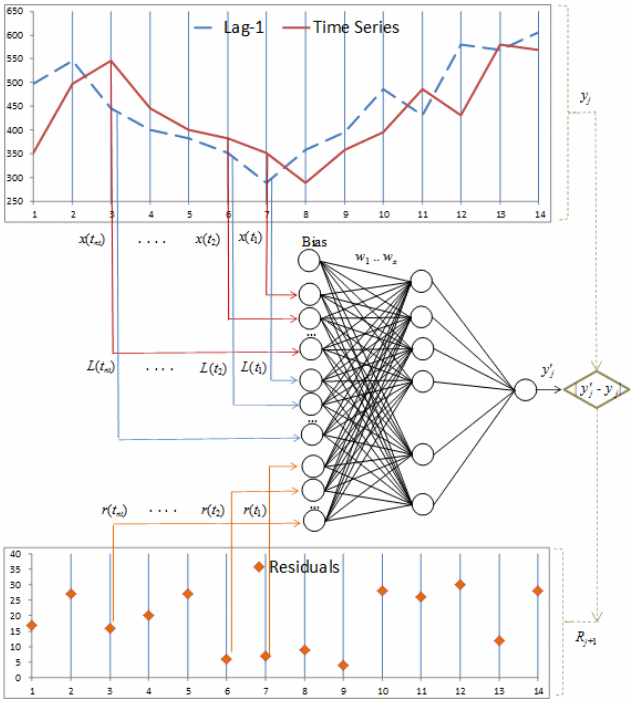


Fig. 3. Additional inputs being fed into PNN-cf network.

PNN+cf is applied here for demonstrating a scenario of using a neural network which can adaptively and dynamically adjust its structure via an optimization process and input sources selection.

III. Experiment

An experiment is carried out to verify the efficacy of the proposed GROOMS, particularly for early stage forecasting of the 2019-nCov epidemic. Empirical data is used, which is extracted from the archive of Chinese health authorities. The data is updated daily since 21 Jan 2020, of the daily increment of the number of infested people in China. The trends are put in categories of which makes up a full cycle of a case after the infection: Confirmed, Cured, Died, Suspected, Critical. A patient is deemed as suspected when s/he started to show symptoms of virus infection; through some medical diagnosis afterward, s/he will be confirmed or clear. The patient then may fall into critical condition which s/he would be either cured or died. These figures show daily trend as the epidemic develops. Till the time of writing, the epidemic trend is still on the rise, though for experiment only the data between 21 Jan – 3 Feb 2020 are used. A time series of 14 instances about the Suspected cases is to be run through GROOMS for 6-days ahead forecast. At any point of time in the progress of the epidemic, authorities are concerned about forecast of future days. The fourteen instances represent a challenging scenario of data mining over small data.

The experimentation platform is a i7-CPU running 64-bits OS of MS Windows 10, with installations of several machine learning benchmarking software – Weka, SPSS, NunXL, XLMiner, etc. from Waikato University, NA, Microsoft Inc, IBM Inc, Spider Software Inc and Solver Inc. respectively. In most cases, default parameters values are used as they are to start with the initial run.

Besides the fact that the curve of Suspected is non-stationary with a rising trend apparently, the technical decomposition is technically challenging – it has a sharp hump and trough near the end of the first week, followed by another smaller reverse trough and hump towards the end of the curve. Intuitively there are no elements of seasonality, cycle, and white noise. Although small the available data amount is, it has partially some but incomplete characteristics of logarithmic curve and repetitive zigzags. From humanity point of view, daily increase of suspected cases might be an important indictor of a wide-spreading epidemic (see Fig. 4).

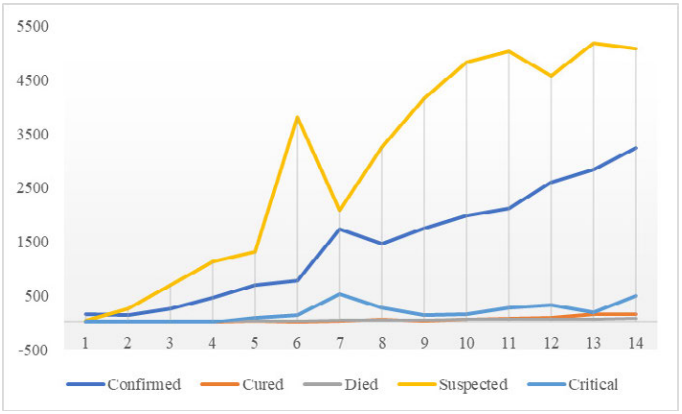


Fig. 4. Daily increase of Wuhan coronavirus infection cases.

Three representative groups of forecasting algorithms are put into testing of GROOMS. They are classical time series forecasting algorithms, machine learning models and polynomial neural networks for time series forecasting respectively. The rational is to observe and investigate how matching the curve fitting produced by these algorithms to the limited epidemic data at the early stage. This is to simulate a situation when the outbreak just occurred, how these forecasting algorithms perform at their best under crisis. Root-mean-square-error

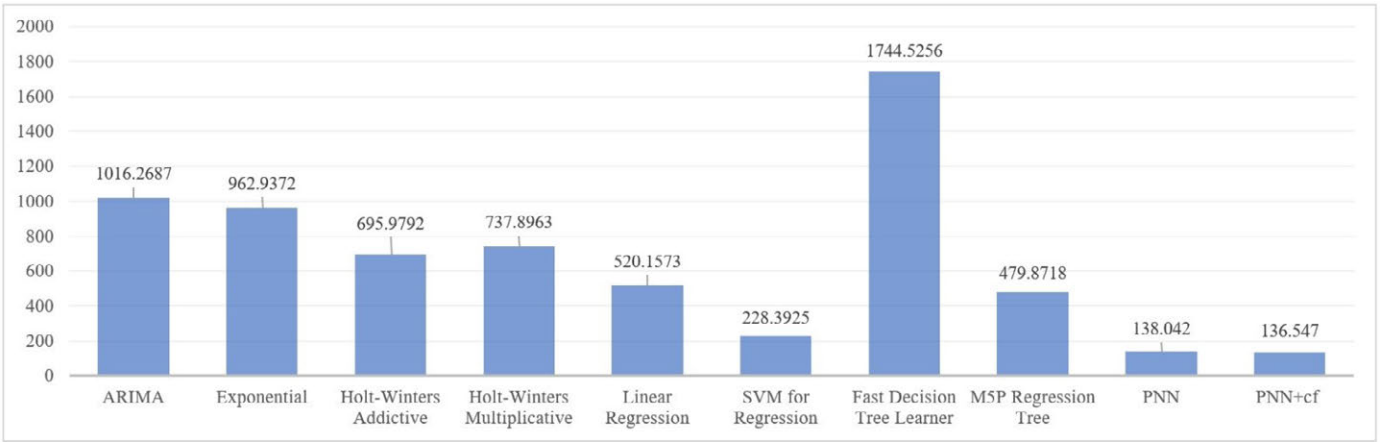
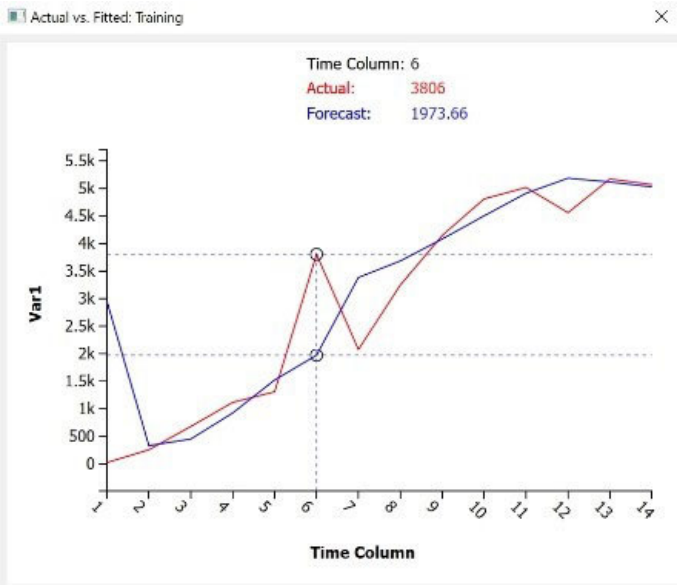


Fig. 5. RMSE comparison between three groups of forecasting methods – traditional (left), machine learning (middle) and PNN (right).

(RMSE) which is a standard benchmarking performance measure in time series forecasting is used as a criterion in panel selection.

The same small dataset is loaded into various above-mentioned algorithms without any pre-processing for competition. If automatic tuning is available for the algorithm, it is activated, or else default parameter values would be assumed. At the end of their final runs, RMSE was recorded down for comparison. They are shown in bar charts as in Fig. 5.

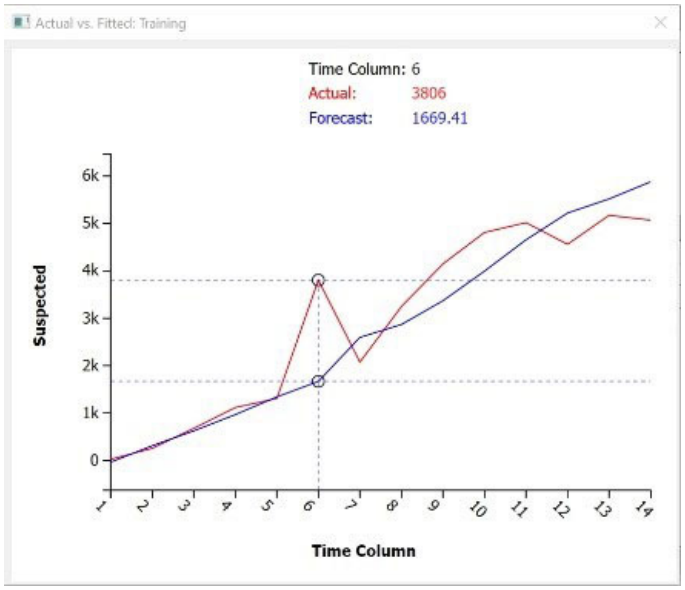
The result bars are positioned in such a way that the classical algorithms are the left, the machine learning algorithms are in the middle, and the neural network-based algorithms are at the right. It can be seen that in general the classical time series forecasting algorithms failed to perform well. There are marginally large gaps mainly between the fitting curve and the actual data at the hump near the initial stage. Please see Fig. 6 a-d.



a) ARIMA

b) Exponential

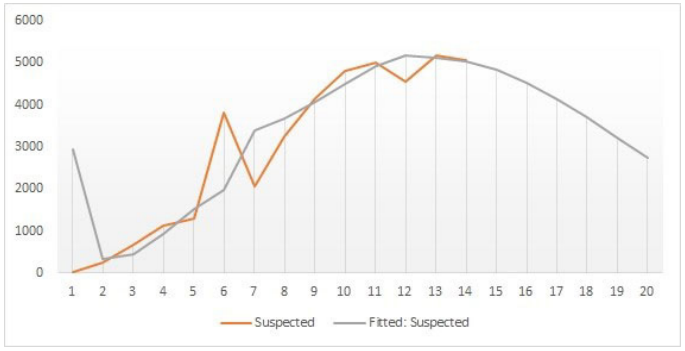
c) Holt-Winters Addictive



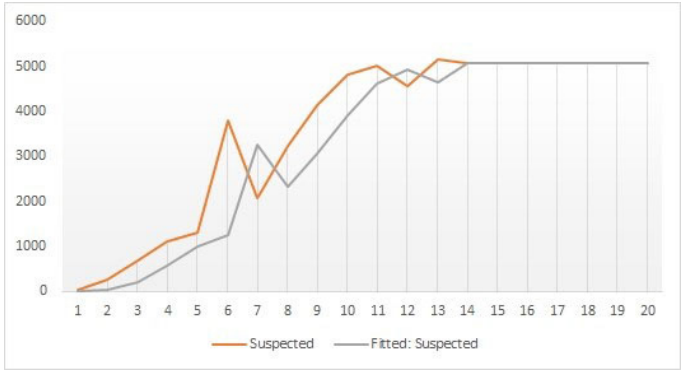
d) Holt-Winters Multiplicative

Fig. 6. Deviation between predicted and actual values.

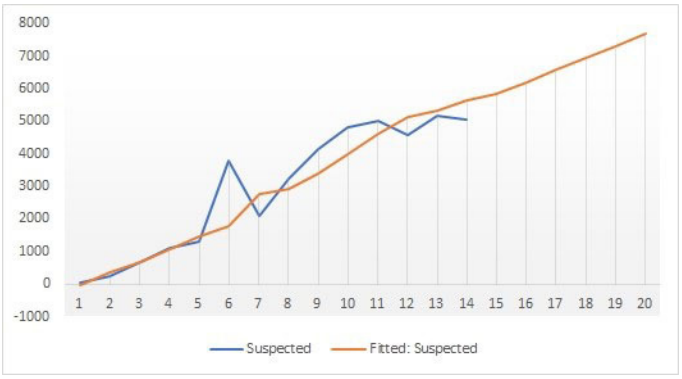
Such drastic changes at the curve hampered the capability of algorithms that are based on curve-fitting. Consequently, by referring to the rates of RMSE, machine learning models appear to be more promising than those classical algorithms for this particular case. An exception is Fast Decision Tree Learner that scored RMSE 1744.526. On the other hand, neural network based PNN's are found to perform the best in this panel selection at the lowest RMSE 136.547 observed. The forecasting results by traditional time series forecasting methods - ARIMA, Exponential, Holt-Winters Addictive and Hot-Winters Multiplicative are shown in Fig. 7 a-d. The forecasts by machine learning models are shown in Fig. 8 a-d.



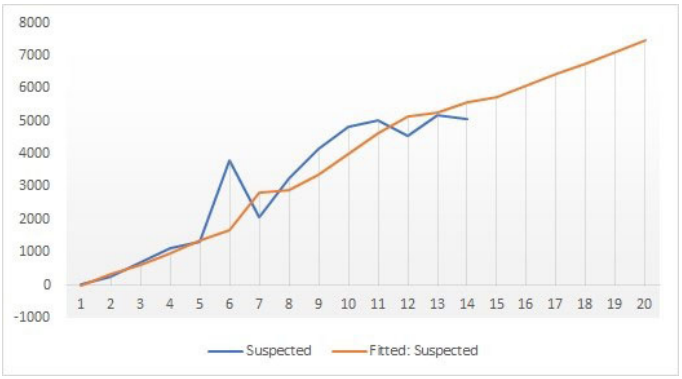
a) ARIMA



b) Exponential



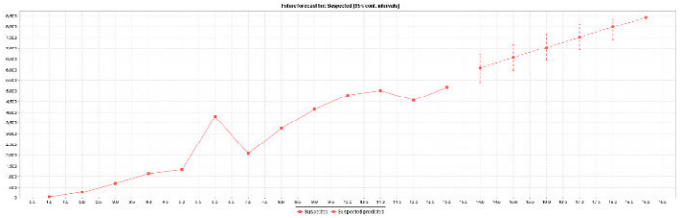
c) Holt-Winters Addictive



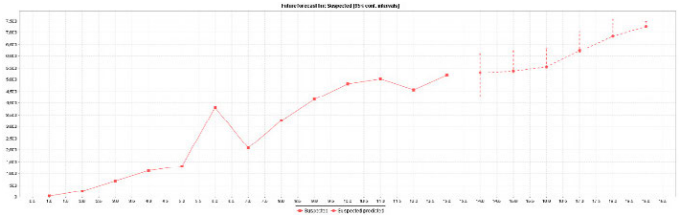
d) Holt-Winters Multiplicative

Fig. 7. Forecasts generated by traditional time series forecasting algorithms.

It is observed that the four forecasting results from Fig. 7 a-d that they basically give very different trends. Holt-Winters' up, ARIMA down, and Exponential flat. Just by judging the RMSE within this group of predictors alone, Holt-Winters has the lowest error. So, it is indicating a rising trend that with an almost constant gradient. However, at the panel selection of GROOMS, there are forecasting models by machine learning methods that have lower errors.



a) Linear Regression



b) SVM for regression [23]

TABLE I. RMSE OF VARIOUS TYPES OF PNN COUPLED WITH DIFFERENT SUBSETS OF INPUT DATA SOURCES

Polynominal NN types	All	suspected + confirmed	suspected + confirmed + critical	suspected + confirmed + critical + cured	suspected + confirmed + critical + died	suspected + critical	suspected + cured	suspected + died
Combi	200.044	190.599*	190.599*	237.261	214.097	193.520	276.885	290.079
Combi-cf	174.366	189.661	160.338*	165.127	172.627	167.544	241.052	222.997
MIA	153.709	155.684	138.042*	141.557	156.275	158.156	183.904	186.212
MIA-cf	151.789	154.128	136.547*	140.780	157.496	162.175	189.954	186.212

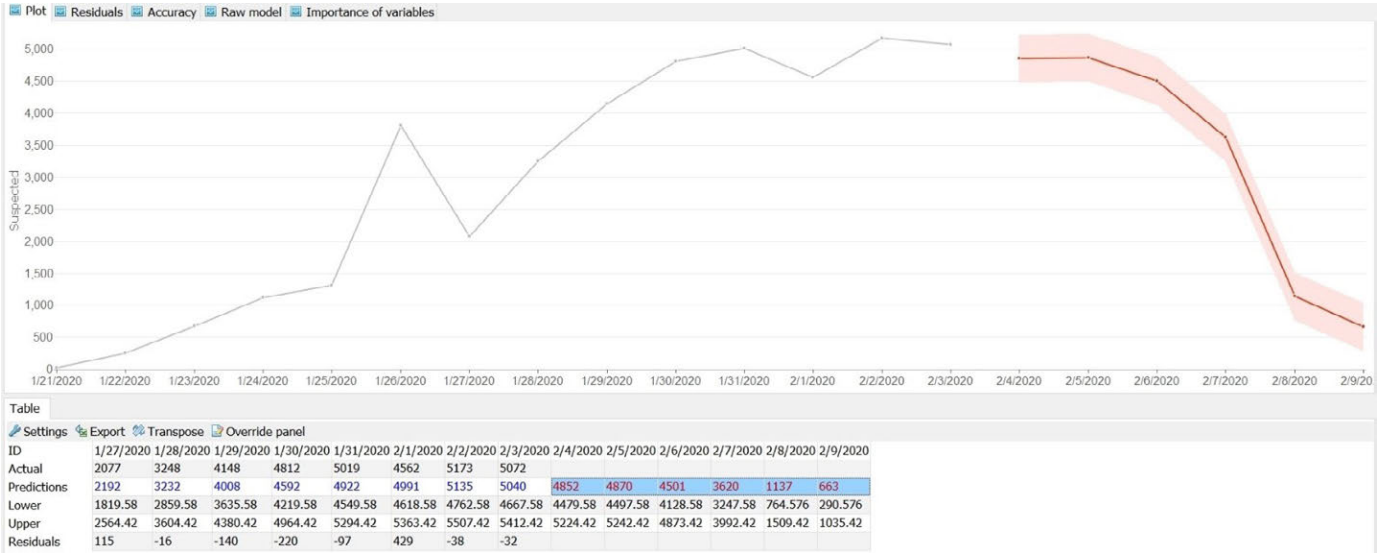
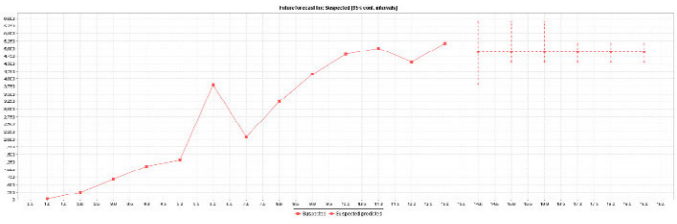


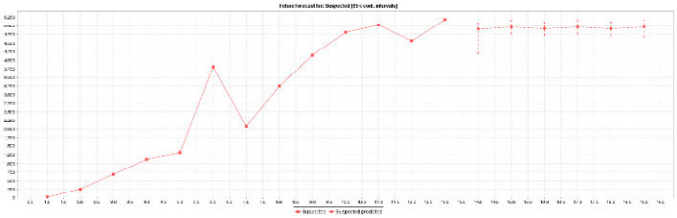
Table
Settings Export Transpose Override panel

ID	1/27/2020	1/28/2020	1/29/2020	1/30/2020	1/31/2020	2/1/2020	2/2/2020	2/3/2020	2/4/2020	2/5/2020	2/6/2020	2/7/2020	2/8/2020	2/9/2020
Actual	2077	3248	4148	4812	5019	4562	5173	5072						
Predictions	2192	3232	4008	4592	4922	4991	5135	5040	4852	4870	4501	3620	1137	663
Lower	1819.58	2859.58	3635.58	4219.58	4549.58	4618.58	4762.58	4667.58	4479.58	4497.58	4128.58	3247.58	764.576	290.576
Upper	2564.42	3604.42	4380.42	4964.42	5294.42	5363.42	5507.42	5412.42	5224.42	5242.42	4873.42	3992.42	1509.42	1035.42
Residuals	115	-16	-140	-220	-97	429	-38	-32						

Fig. 9. Forecasts generated by PNN+cf algorithms.



c) Fast decision tree learner [24]



d) MP5 decision tree leaner [25]

Fig. 8. Forecasts generated by machine learning algorithms.

The forecasts by this group which are shown in Fig. 8 appear like straight lines except SVM for Regression. This suggest that SVM which is a strong hyperplane classifier capable of recognizing very non-linear pattern (in this case, the epidemic trend is indeed curvy and non-smooth), it is able to produce a forecast with certain blends and turns, based on the past history which is anything but smooth.

The winning model from panel selection is PNN+cf which evolved its structure to its most optimal. It belongs to Group 1 according to GROOMS, which allows inclusion and selection of multiple data sources other than the one that is being forecasted. A separate experiment is therefore conducted that tried several combinations of data sources; from there we observe which combo has the lowest error rates. Please refer to the results in Table I. By default, when all data sources are used together in the multiple regression, mediocre performance are obtained. It was found that, for both types of Combi and MIA architectures of PNN, this specific combination of data sources - suspected + confirmed + critical, when used together, superior performances are obtained. Intuitively, this is a natural sequence that resembles the lifecycle of an infested patient. An infection case that contributed to the national health records, started with somebody who displayed certain symptoms, hence a suspected case. After some diagnosis and test, the disease can be confirmed as coronavirus infection or something else. The patient received medical treatment; he may then turn critical that probably lead to fatality or cured. So, in order to forecast future suspected cases, the closely related status in sequence, that are confirmed and critical, could help. These three statuses form some kind of causality. They exist as a prerequisite for one to another to occur with high certainty. However, there exists a large uncertainty beyond the condition of critical at the moment, given that the virus is novel, its severity is not well-known, and the amount of available data is scarce. By the predicting powers of neural network, this uncertainty is reflected by the relatively higher RMSE on the right side of Table I.

Since the winning model PNN+cf offers the lowest RMSE among all, its forecasting result is shown in Fig. 9. If the data does not change (theoretically), the trend is forecast to dip down and the epidemic subsides. Of course, it is only a forecast as accurate as up to the current point of time, that is up to 4 Feb 2020. In this dynamic situation, the latest figures keep being updated daily, so does the forecast. Whenever new data comes in, it is highly likely that it will change the course of prediction, and a totally new outcome will be obtained which may be very different from the previous one. The future would unfold itself when time progresses.

IV. Conclusion

Forecasting the ending time of an epidemic is always a noble but difficult task. From the perspective of data mining, the quest of finding the most accurate machine learning model while limited with very few data on hand to start about has attracted research attentions. In literature there are many bootstrapping and data augmentation methods for improving the performance of machine learning models. In this paper, a methodology called GROOMS is proposed which ensembles a collection of five types of forecasting methods, ranging from classical time-series forecasting, to self-evolving polynomial neural networks (PNN), to work together under a panel selector. In particular, our experiment has shown that PNN is superior in yielding a forecasting model with relatively lowest error. PNN is known to possess advantages in forming up a just fit model by incrementally growing its internal structure from simple to optimal. Therefore, over- or under-fitting is less likely to happen. In this paper, PNN is extended to PNN+cf with corrective feedback in the optimization process. At the best efforts in optimizing the parameters, input data sources for multiple regression using GROOMS, it is shown the possibility and feasibility in finding the most accurate forecasting model based on the limited availability of data. Although the experiment demonstrated the possibility of picking a suitable forecasting model, forecasting is a dynamic task itself whose predicted result is very sensitive to the parameters used, the choice of model and the training data. Any addition or modification on the training data upon the arrival of new data, is likely to influence the course of the predicted result (as seen in Fig. 7 – the uptrend vs downtrend).

In future, the underlying cause of the differences or deviation of forecasting results by different algorithms worth in-depth investigation. From data mining viewpoint, it is vital to test more other popular algorithms for panel selection. Better still, it worth the efforts in probing in the underlying designs of some prominent algorithms and understanding why or why not it incur a very low or very high errors. With this insight discovered, one may formulate even new breed of algorithms from inside out, which handles well small data yet giving acceptable quality forecast. The methodology should be extended too. So far GROOMS was constructed in mind that the accuracy/error level is the sole criterion. In reality, pondering whether an epidemic is going to end or escalate is a complex and wide multi-factors decision. Based solely on historical data is only of one aspect of contributions to the decision, though it is still important. A more thorough methodology would be useful for connecting the technical forecast to other non-technical decision-making processes. So, they can complement each other for making a multi-facet and reliable decision.

References

[1] "WHO | Novel Coronavirus – China". WHO. Archived from the original on 23 January 2020. Retrieved 1 February 2020.

[2] Cohen, Jon (January 2020). "Wuhan seafood market may not be source of novel virus spreading globally". Science. doi:10.1126/science.abb0611. ISSN 0036-8075.

[3] "Statement on the second meeting of the International Health Regulations (2005) Emergency Committee regarding the outbreak of novel coronavirus (2019-nCoV)". World Health Organization (WHO). 30 January 2020. Archived from the original on 31 January 2020. Retrieved 30 January 2020.

[4] Sparrow, Annie. "How China's Coronavirus Is Spreading—and How to Stop It". Foreign Policy. Archived from the original on 31 January 2020. Retrieved 2 February 2020.

[5] Croda, R. M. C., D. E. G. Romero, and S. O. C. Morales, "Sales Prediction through Neural Networks for a Small Dataset", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 4, pp. 35-41, 03/2019.

[6] R. J. Hyndman, and A. V. Kostenko, "Minimum sample size requirements for seasonal forecasting models," Foresight, vol. 6, pp. 12-15, 2007.

[7] S. Ingrassia, and I. Morlini, "Neural network modeling for small datasets," Technometrics, vol.47, no. 3, pp. 297-311, 2005.

[8] A. Pasini, "Artificial neural networks for small dataset analysis." Journal of thoracic disease, vol. 7, no. 5, pp- 953, 2015.

[9] M. A. Lateh, A. K. Muda, Z. I. M. Yusof, N. A. Muda and M. S. Azmi, "Handling a Small Dataset Problem in Prediction Model by employ Artificial Data Generation Approach: A Review", Journal of Physics: Conference Series, Volume 892, Conf. Ser. 892 012016.

[10] R. Andonie, "Extreme data mining: Inference from small datasets," Int. J. Comput. Commun. Control, vol. 5, no. 3, pp. 280–291, 2010.

[11] T. Shaikhina, N. A.Khovanova, "Handling limited datasets with neural networks in medical applications: A small-data approach", Artificial Intelligence in Medicine, vol. 75, pp. 51-63, 2017.

[12] J. F. Slifker and S. S. Shapiro, "The Johnson system: selection and parameter estimation," Technometrics, vol. 22, no. 2, pp. 239–246, 1980

[13] R. Adhikari, and R. K. Agrawal, "An introductory study on time series modeling and forecasting," arXiv preprint arXiv:1302.6613, 2013.

[14] A. Singh, and G. C. Mishra, "Application of Box-Jenkins method and Artificial Neural Network procedure for time series forecasting of prices," Statistics in Transition new series, vol. 1, no. 16, pp. 83-96, 2015.

[15] Arun, V., M. Krishna, B. V. Arunkumar, S. K. Padma, and V. Shyam, "Exploratory Boosted Feature Selection and Neural Network Framework for Depression Classification", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 3, pp. 61-71, 12/2018

[16] A. G. Ivakhnenko (1970) "Heuristic Self-Organization in Problems of Engineering Cybernetics". Automatica Vol. 6, pp.207–219.

[17] A. G. Ivakhnenko, and A. A. Zholnarskiy, (1992) "Estimating the coefficients of polynomials in parametric GMDH algorithms by the improved instrumental variables method", Journal of Automation and Information Sciences c/c of Avtomatika, Vol. 25, no. 3, pp.25-32.

[18] S. K. Oh, W. Pedrycz, B. J. Park, "Polynomial neural networks architecture: analysis and design", Computers & Electrical Engineering, Vol. 29, No. 6, August 2003, pp. 703-725.

[19] A. Andoni, R. Panigrahy, G. Valiant, L. Zhang, "Learning Polynomials with Neural Networks", Proceedings of the 31 st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

[20] S. Fong, N. N. Zhou, R. K. Wong, X. S. Yang: Rare Events Forecasting Using a Residual-Feedback GMDH Neural Network. ICDM Workshops 2012: 464-473.

[21] N. Dey, S. Fong, W. Song, K. Cho (2018) Forecasting Energy Consumption from Smart Home Sensor Network by Deep Learning. In: Deshpande A. et al. (eds) Smart Trends in Information Technology and Computer Communications. SmartCom 2017. Communications in Computer and Information Science, vol 876. Springer.

[22] W. Waheeb, and R. Ghazali, Forecasting the Behavior of Gas Furnace Multivariate Time Series Using Ridge Polynomial Based Neural Network Models, International Journal of Interactive Multimedia and Artificial Intelligence, ISSN 1989-1660, vol. 5, no. 5, 06/2019, pp.126-133.

[23] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, K.R.K. Murthy: Improvements to the SMO Algorithm for SVM Regression. In: IEEE Transactions on Neural Networks, 1999.

[24] J. Su, H. Zhang, A fast decision tree learning algorithm, AAAI'06: Proceedings of the 21st national conference on Artificial intelligence – Vol. 1, July 2006, pp.500–505.

[25] Y. Wang, I. H. Witten: Induction of model trees for predicting continuous classes. In: Poster papers of the 9th European Conference on Machine Learning, 1997.

Simon James Fong

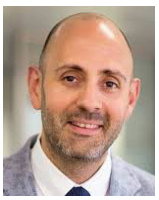
Simon Fong graduated from La Trobe University, Australia, with a 1st Class Honours BEng. Computer Systems degree and a PhD. Computer Science degree in 1993 and 1998 respectively. Simon is now working as an Associate Professor at the Computer and Information Science Department of the University of Macau, as an Adjunct Professor at Faculty of Informatics, Durban University of Technology, South Africa. He is a co-founder of the Data Analytics and Collaborative Computing Research Group in the Faculty of Science and Technology. Prior to his academic career, Simon took up various managerial and technical posts, such as systems engineer, IT consultant and e-commerce director in Australia and Asia. Dr. Fong has published over 450 international conference and peer-reviewed journal papers, mostly in the areas of data mining, data stream mining, big data analytics, meta-heuristics optimization algorithms, and their applications. He serves on the editorial boards of the Journal of Network and Computer Applications of Elsevier, IEEE IT Professional Magazine, and various special issues of SCIE-indexed journals. Simon is also an active researcher with leading positions such as Vice-chair of IEEE Computational Intelligence Society (CIS) Task Force on "Business Intelligence & Knowledge Management", TC Chair of IEEE ComSoc e-Health SIG and Vice-director of International Consortium for Optimization and Modelling in Science and Industry (iCOMSI).

Gloria Li

Gloria Tengyue Li is currently a PhD student at the University of Macau. She is also the Head of Data Analytics and Collaborative Computing Laboratory, Zhuhai Institute of Advanced Technology, Chinese Academy of Science, Zhuhai, China. Ms Li is leading and managing the laboratory, in R&D as well as technological transfer and incubation. She is an entrepreneur with experiences in innovative I.T. contest, with her award-winning team in the Bank of China Million Dollar Cup competition. Her latest winning work includes the first unmanned supermarket in Macau enabled by the latest sensing technologies, face recognition and e-payment systems. She is also the founder of several Online2Offline dot.com companies in trading and retailing both online and offline. Ms Li is also an active researcher, manager and chief-knowledge-officer in DACC laboratory at the faculty of science and technology, University of Macau.

Nilanjan Dey

Nilanjan Dey, is an Assistant Professor in Department of Information Technology at Techno International New Town (Formerly known as Techno India College of Technology), Kolkata, India. He is a visiting fellow of the University of Reading, UK. He was an honorary Visiting Scientist at Global Biomedical Technologies Inc., CA, USA (2012-2015). He was awarded his PhD. from Jadavpur University in 2015. He is the Editor-in-Chief of International Journal of Ambient Computing and Intelligence, IGI Global. He is the Series Co-Editor of Springer Tracts in Nature-Inspired Computing, Springer Nature, Series Co-Editor of Advances in Ubiquitous Sensing Applications for Healthcare, Elsevier, Series Editor of Computational Intelligence in Engineering Problem Solving and Intelligent Signal processing and data analysis, CRC. He has authored/edited more than 50 books with Springer, Elsevier, Wiley, CRC Press and published more than 300 peer-reviewed research papers. His main research interests include Medical Imaging, Machine learning, Computer Aided Diagnosis, Data Mining etc. He is the Indian Ambassador of International Federation for Information Processing (IFIP) – Young ICT Group.



Rubén González Crespo

Dr. Rubén González Crespo has a PhD in Computer Science Engineering. Currently he is Vice Chancellor of Academic Affairs and Faculty from UNIR and Global Director of Engineering Schools from PROEDUCA Group. He is advisory board member for the Ministry of Education at Colombia and evaluator from the National Agency for Quality Evaluation and Accreditation of Spain (ANECA). He is member from different committees at ISO Organization. Finally, He has published more than 200 papers in indexed journals and congresses.

Enrique Herrera-Viedma

Enrique Herrera-Viedma is Professor in Computer Science and A.I in University of Granada and currently the new Vice-President for Research and Knowledge Transfer. His current research interests include group decision making, consensus models, linguistic modeling, and aggregation of information, information retrieval, bibliometric, digital libraries, web quality evaluation, recommender systems, and social media. In these topics he has published more than 250 papers in ISI journals and coordinated more than 22 research projects. Dr. Herrera-Viedma is Vice-President of Publications of the IEEE SMC Society and an Associate Editor of international journals such as the IEEE Trans. On Syst. Man, and Cyb.: Systems, Knowledge Based Systems, Soft Computing, Fuzzy Optimization and Decision Making, Applied Soft Computing, Journal of Intelligent and Fuzzy Systems, and Information Sciences.

LA UNIVERSIDAD
EN INTERNET

Rectorado
Avenida de la Paz, 137
26006 Logroño (La Rioja)
t (+34) 941 21 02 11

www.unir.net