In the Name of God

Journal of
Information Systems & Telecommunication

Vol. 9, No.2, April-June 2021, Serial Number 34

Research Institute for Information and Communication Technology
Iranian Association of Information and Communication Technology
Affiliated to: Academic Center for Education, Culture and Research (ACECR)

Manager-in-Charge: Habibollah Asghari, ACECR, Iran
Editor-in-Chief: Masoud Shafiee, Amir Kabir University of Technology, Iran

Guest Editor: Dr. Omid Mahdi Ebadati
Executive Editor: Dr. Fatemeh Kheirkhah
Executive Manager: Shirin Gilaki
Executive Assistants: Mahdokht Ghahari, Ali Mokhtarani
Print ISSN: 2322-1437
Online ISSN: 2345-2773
Publication License: 91/13216
Editorial Office Address: No.5, Saeedi Alley, Kalej Intersection., Enghelab Ave., Tehran, Iran,
P.O.Box: 13145-799 Tel: (+9821) 88930150 Fax: (+9821) 88930157
E-mail: info@jist.ir , infojist@gmail.com
URL: www.jist.ir

Indexed by:

- SCOPUS	www. Scopus.com
- Index Copernicus International	www.indexcopernicus.com
- Islamic World Science Citation Center (ISC)	www.isc.gov.ir
- Directory of open Access Journals	www.Doaj.org
- Scientific Information Database (SID)	www.sid.ir
- Regional Information Center for Science and Technology (RICeST)	www.ricest.ac.ir
- Iranian Magazines Databases	www.magiran.com

Publisher:
Iranian Academic Center for Education, Culture and Research **(ACECR)**

This Journal is published under scientific support of
Advanced Information Systems (AIS) Research Group and
Digital & Signal Processing Research Group, ICTRC

Acknowledgement

JIST Editorial-Board would like to gratefully appreciate the following distinguished referees for spending their valuable time and expertise in reviewing the manuscripts and their constructive suggestions, which had a great impact on the enhancement of this issue of the JIST Journal.

(A-Z)

- Agahi, Hamed, Islamic Azad University of Shiraz, Iran
- Agarwal, Parul, Jamia Hamdard University, New Delhi, India
- Ahmadizad, Arman, University of Kurdistan, Kurdistan, Iran
- Alavi, Seyed Enayatallah, Shahid Chamran University, Ahvaz, Iran
- Alam Tabriz, Akbar, Shahid Beheshti University, Tehran, Iran
- Azizi, Sadoon, University of Kurdistan, Sanandaj, Iran
- Badie, Kambiz, Tehran University, Tehran, Iran
- Babaei, Shahram, Islamic Azad University of Tabriz, Iran
- Bahrepour, Davoud, Islamic Azad University, Mashhad, Iran
- Bayanati, mahmonir, West Tehran Islamic Azad University, Tehran, Iran
- Bhat, Mudasir Ashraf, University of Kashmir, Srinagar, Kashmir
- Bouyer, Asgar Ali, Shahid Madani University, Azarbaijan, Iran
- Darmani, Yousef, K. N. Toosi University of Technology, Tehran, Iran
- Davarpanah, seyed hashem, Birjand University of Technology, South Khorasan, Iran
- Ebadati, Omid Mahdi, Kharazmi University,Tehran, Iran
- Ebrahimpour, Nader, Islamic Azad University of Mahabad, West Azerbaijan, Iran
- Farsi, Hassan, University of Birjand, South Khorasan, Iran
- Fatemi Khorasgani, Afsaneh, University of Isfahan, Isfahan, Iran
- Fekri Ershad, Shervan, Islamic Azad University, Najafabad, Iran
- Ghayoomi, Masood, Institute for Humanities and Cultural Studies, Tehran, Iran
- Ghaffari, Ali, Islamic Azad University, Tabriz Branch, Iran
- Ghasemzadeh, Mohammad, Yazd University, Yazd, Iran
- Haghighi, Hassan, Shahid Beheshti University, Tehran, Iran
- Kheirkhah, Fatemeh, ACECR, Tehran, Iran
- Mavaddati, Samira, University of Mazandaran, Iran
- Mirroshandel, Seyed Abolghasem, University of Guilan, Rasht, Iran
- Mirzaei, Abbas, Islamic Azad University, Ardabil, Iran
- Minoofam, Seyed Amir Hadi, Qazvin Islamic Azad University, Qazvin, Iran
- Mnkandla, Enerst, University of South Africa, Africa
- Moussaoui, Abdelkrim, Guelma University, Algeria
- Mohammadzadeh, Sajjad, University of Birjand, South Khorasan, Iran
- Momtazi, Saeedeh, Amir kabir University, Tehran, Iran
- Mohammadi, Mohammad Reza, Iran University of Science and Technology, Tehran, Iran
- Mothku, Sai Krishna, National Institute of Technology Tiruchirappalli, Tamil Nadu, India

- Nangir, Mahdi, University of Tabriz, Tabriz, Iran
- Napitupulu, Darmawan, Indonesian Institute of Sciences, Jakarta, Indonesia
- Paindavoine, Michel, University of Bourgogne, Dijon, France
- Pirgazi, Jamshid, University of Zanjan, Zanjan, Iran
- Reshadat, Vahideh, Malek-Ashtar University of Technology, Tehran, Iran
- Sarrafzadeh, Abdolhossein, Unitec Institute of Technology, Auckland, New Zealand
- Sekar, M, Raja, VNR VJIET, Institute of Engineering and Technology, Hyderabad, India
- Shirmarz, Alireza, Ale Taha Institute of Higher Education, Tehran, Iran
- Sohrabi, Babak, Tehran University, Tehran, Iran
- Solouk, Vahid, Urmia University of technology, Iran
- Vu Duc, Nghia, Chung-Ang University, Seoul, Korea
- Yaghoobi, Kaebeh, Ale Taha Institute of Higher Education, Tehran, Iran
- Zare, Hadi, Tehran University, Tehran, Iran

Table of Contents

Performance Analysis of Hybrid SOM and AdaBoost Classifiers for Diagnosis of Hypertensive Retinopathy

Wiharto*
Department of Informatics, Universitas Sebelas Maret, Surakarta, Indonesia
wiharto@staff.uns.ac.id
Esti Suryani
Department of Informatics, Universitas Sebelas Maret, Surakarta, Indonesia
estisuryani@staff.uns.ac.id
Murdoko Susilo
Department of Informatics, Universitas Sebelas Maret, Surakarta, Indonesia
murdokosusilo@student.uns.ac.id

Received: 17/Jan/2021 Revised: 26/Mar/2021 Accepted: 21/Apr/2021

Abstract

The diagnosis of hypertensive retinopathy (CAD-RH) can be made by observing the tortuosity of the retinal vessels. Tortuosity is a feature that is able to show the characteristics of normal or abnormal blood vessels. This study aims to analyze the performance of the CAD-RH system based on feature extraction tortuosity of retinal blood vessels. This study uses a segmentation method based on clustering self-organizing maps (SOM) combined with feature extraction, feature selection, and the ensemble Adaptive Boosting (AdaBoost) classification algorithm. Feature extraction was performed using fractal analysis with the box-counting method, lacunarity with the gliding box method, and invariant moment. Feature selection is done by using the information gain method, to rank all the features that are produced, furthermore, it is selected by referring to the gain value. The best system performance is generated in the number of clusters 2 with fractal dimension, lacunarity with box size 2^2-2^9, and invariant moment M_1 and M_3. Performance in these conditions is able to provide 84% sensitivity, 88% specificity, 7.0 likelihood ratio positive (LR+), and 86% area under the curve (AUC). This model is also better than a number of ensemble algorithms, such as bagging and random forest. Referring to these results, it can be concluded that the use of this model can be an alternative to CAD-RH, where the resulting performance is in a good category.

Keywords: Hypertensive Retinopathy; Self-organizing Maps; Segmentation; Adaboost; Classification; Information Gain.

1- Introduction

Hypertension can be detected when we are diligent in checking blood pressure. Hypertension can cause severe health complications and increase the risk of heart disease, stroke, and sometimes death. Hypertension can also cause damage to the retina and blood vessels around the retina, a condition called hypertensive retinopathy. In hypertensive retinopathy, there is the thickening of the blood vessels, which in turn can disrupt blood flow to the retina. Disruption of blood flow to the retina can cause vision problems.

Hypertensive retinopathy can be detected by analyzing the retina of the eye. Analysis can be carried out in person by the clinician or with the aid of a computer. Analysis for diagnosis is carried out with the help of computers, namely processing the retinal image from the fundus camera. The hypertensive retinopathy diagnosis model generally has preprocessing, segmentation, feature extraction, classification, and performance analysis stages[1].

An important stage in the diagnosis process is the segmentation, feature extraction, and classification stages. These three stages have many methods used, such as the segmentation stage. Segmentation has several approaches, one of which is clustering [2]. Retinal image segmentation has been done a lot, as done by Wiharto et al.[3]. This study analyzes the effect of the number of clusters on segmentation performance using the fuzzy c-means clustering algorithm. Besides, this study also conducted a comparison of the mean and median methods in determining the threshold used to separate blood vessels from the background. The same thing was done by Wiharto et al.[4], namely segmentation using the k-means algorithm, with the method of determining the threshold using the mean of the center of the cluster.

The clustering approach used in the blood vessel segmentation process does not only feature clustering-based but also Neural Network-based [2]. The neural network-based use includes self-organizing maps (SOM),

* Corresponding Author

as used in the research of Wiharto et al.[5]. In this study, retinal blood vessels were segmented using SOM combined with the determination of the threshold which was the median of the center of the cluster. The use of clustering for segmentation was also used in the study of Shafiei et al.[6] for the case of CT scan images for the detection of lung cancer tumors. In this study, the algorithm for segmentation is SLIC (simple linear iterative clustering) based on k-means clustering. Referring to the study of Lupascu et al.[7], explained that SOM is better than k-means for retinal vascular segmentation. The ability to perform SOM clustering is also described in the research of Budayan et al.[8], and in general image segmentation with SOM and FCM is better than k-means [9]–[11].

The next stage in CAD-RH is feature extraction. Referring to the segmentation stage, which mostly focuses on the segmentation of retinal blood vessels, the feature extraction stage uses several methods such as fractal dimensions and lacunarity. The fractal and lacunarity dimensions have been associated with hypertension, arterial, and venous blood vessels in the retina [12]–[15]. Referring to this, a number of studies have used feature extraction for CAD-RH. Research conducted by Wiharto et al. [16] used the fractal and lacunarity dimensions, where the fractal dimensions used the box-counting method. The fractal dimension was also used in a study conducted by Syahputra et al. [17] but in combination with the invariant moment. Both studies use segmentation with a threshold approach. Invariant moments were also used in the study of Narasimhan et al.[18], but combined with gray levels. The same feature extraction as Syahputra et al. [17] was also used by Hutson et al.[19] but in the case of CAD-Diabetic Retinopathy.

The last stage in CAD-RH after feature selection is classification. Classification methods that have been used in the CAD-RH model are classification based on artificial neural networks, decision trees, naïve Bayesian, support vector machines, and ensemble learning [16]–[18], [20], [21]. These methods have a number of drawbacks, one of which is overfitting. The method that can overcome overfit is the ensemble method. The ensemble method has a number of algorithms such as random forest and AdaBoost. The AdaBoost algorithms has better ability than random forest [22] and has better overfit ability [23].

Referring to studies that have been carried out in the segmentation and classification stages, it shows that the segmentation performance using the clustering approach is able to provide performance with an excellent AUC value above 90%, namely the SOM method. The ability of SOM is better than segmentation by using a combination of frangi filter and otsu thresholding. The results of segmentation with SOM have not been tested whether they are able to produce features that can be optimally used for CAD-RH. Feature extraction used in a number of previous studies includes fractal, lacunarity, and invariant moment dimensions, with the segmentation method using a threshold-based approach. Another thing from previous studies at the classification stage, most of them use algorithms that are not able to overcome overfit.

Referring to a number of studies that have been carried out, this study analyzes the performance of the CAD-RH system, where the segmentation is clustering-based, namely the SOM. Feature extraction uses the dimensions of fractal, lacunarity, and invariant moment, because of the large number of features it is necessary to scale to select the features used in the classification. The hoisting method used is information gain. The classification algorithm uses AdaBoost. CAD-RH performance is measured using parameters of accuracy, sensitivity, specificity, area under the curve (AUC), likelihood ratio positive (LR+), and likelihood ratio negative (LR-).

2- Material and Method

2-1- Dataset

This research method uses a number of stages which can be shown in Figure 1. The core stages in this research are divided into 6 stages, namely preprocessing, segmentation, feature extraction, feature selection, classification, and performance analysis. This study uses data obtained online, namely the STARE (Structured Analysis of the Retina) dataset. The dataset consists of 50 data, with the distribution of 25 healthy retinal data and 25 retinal data identified hypertensive retinopathy.

2-2- Preprocessing

Retinal image preprocessing was performed to overcome noise, poor contrast, and irregular blood vessel width[24]. The image that is obtained in the dataset is a color image, for that it needs to be converted into a gray image. This is based on research by Dey et al.[24] and Kande et al.[25], who changed the retinal image to a gray image first before segmenting it by clustering. Before the retinal image is further processed, it is converted into 3 channels of the gray image. The gray image taken is the green channel. Green canal image yields significant information about blood vessels and retinal structures because it has the best light reflection [26]. The use of the preprocessing stage is very important. If preprocessing is not used, the segmentation process of blood vessels will result in low segmentation quality. The low quality is due to the number of pixels of the blood vessels that are translated as background, so it will have an impact on the diagnosis result.

The next stage of the green channel image is changed to a negative intensity or it is also called an inverse operation, then the CLAHE process is carried out on the negative image. This is intended to highlight the characteristics of blood vessels. The process of removing the optical disk was carried out using the opening morphology with a structure ball element, which the structure is measuring at 17x17 against the CLAHE result image. The next step is to subtract the CLAHE result image with the opening morphology image, then the optical disk removal image is obtained.

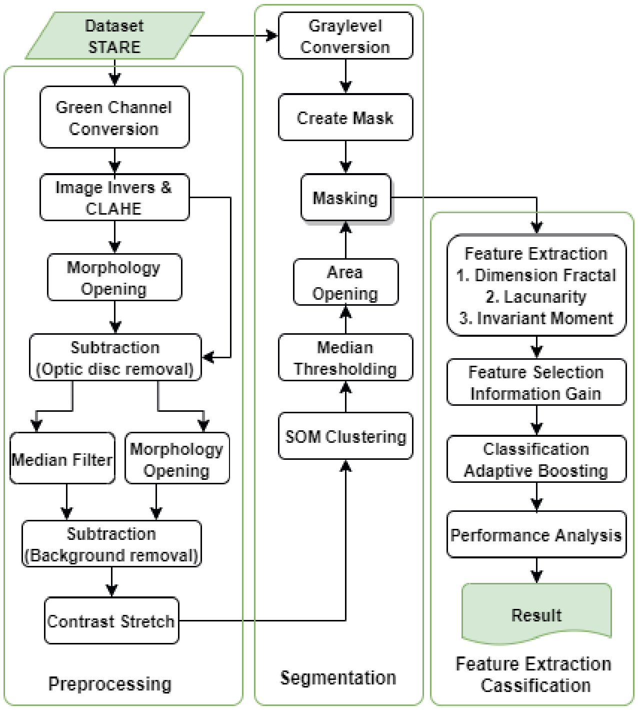


Fig. 1: Research Method

The next process is background subtraction. The image resulting from optical disk removal was processed using a 3x3 median filter, then the morphology of the opening was performed with a 29x29 size disk structure element in the median filter image. The image resulting from the median filter will be subtracted with the image resulting from the opening morphology, then the image resulting from the background subtraction is obtained. The background subtraction process will make the background darker and the veins more prominent, as well as smooth the image texture. However, background subtraction causes the image to appear darker, this is because the subtraction between pixels makes the pixel value decrease. To increase the brightness, the image contrast is increased. Contrast enhancement is done by the contrast stretching process.

2-3- Segmentation

After preprocessing, the segmentation process is carried out using the SOM clustering method. The parameters used in SOM are neighborhood 3, 200 iterations. SOM clustering of retinal images will produce a cluster of k cluster centers. Each cluster has a centroid, in the case of SOM, the centroid of a cluster is the weight of the neuron. The next step is to obtain the image of the blood vessels by thresholding the contrast stretched image. The thresholding process is carried out using the median centroid value of the cluster generated in the SOM process [3], [5]. The result of thresholding is a binary image. The next step is to process the opening area with a radius of 30 to remove small areas in the binary image.

The last step in the segmentation stage is to combine the resulting image of the opening area with the image that was the result of the masking, by multiplying each pixel. This process is done to remove the cover on the retinal image. The process of making a masking image is done by changing the retina image to grayscale, then every pixel that has a value of more than 45 is converted to 1 and the others are converted to 0.

2-4- Feature Extraction

The CAD-RH stage after segmentation is feature extraction. Feature extraction is done by processing the resulting image from segmentation using the fractal dimension, lacunarity, and invariant moment method to obtain image features. Fractal-based feature extraction is used to identify retinal vascular patterns. One of the signs of hypertensive retinopathy is the appearance of tortuosity in the blood vessels that will affect the pattern of the blood vessels. It can be analyzed using fractal analysis, both dimensions and lacunarity [27]. In order to strengthen the features, an invariant moment is added, to see the features from the shape side. This method has the ability not to be susceptible to image changes caused by Rotation, Scale, and Translation [28].

2-4-1- Fractal Dimension

Fractal is a simple geometry that can be broken in such a way that it becomes several parts that have the previous shape with a smaller size [29]. This study uses the box-counting method to calculate the fractal dimensions of an image. Box-counting is done by dividing the image into smaller squares of a certain size.

The following are the steps for the Box Counting method according to Backes and Bruno [30] :

a. The image is divided into squares of size r. The value of r is changed from 1 to 2k, where k = 0, 1, 2, ... and so on, the value from 2k cannot exceed the image size.
b. Counts the number of N boxes containing parts of the object in the image. The value of N is very dependent on r.
c. Calculates the value of log(1/r) and log(N).
d. Make a straight line using log(1/r) and log(N) values.

e. Calculate the slope (slope) of a straight line with equations

$$\propto = \frac{\left(\sum_{k=1}^{n} xy\right) - \frac{\left(\sum_{k=1}^{n} x\right)\left(\sum_{k=1}^{n} y\right)}{n}}{\left(\sum_{k=1}^{n} x^2\right) - \frac{\left(\sum_{k=1}^{n} x\right)^2}{n}} \quad (1)$$

This slope value is the fractal dimension of the image (FD) with the equation:

$$\mathrm{FD} = -\propto \quad (2)$$

Where ∝ is the slope, n is the number of data used, x is the log(1/r) value and y is the log(N) value.

2-4-2- Lacunarity

One of the methods used to calculate lacunarity is the Gliding Box by Allain and Cloitre [31]. This method using a box with the size r x r, to calculate the lacunarity value by recording the pixel value in the box. The Gliding Box steps start by placing an r x r rectangle in the top left corner of an image. The box will process each pixel that contains 1 or 0 until all pixels are passed by the box. The distribution frequency of the pixel contents in each box is denoted by n (M, r). This frequency distribution will be used to determine the probability distribution of each value in the box, denoted by Q (M, r). This probability distribution is obtained from dividing the distribution per pixel by the maximum total number of runs of the box, which is denoted N (r). then the two distributions will be processed with the formula.

$$\mathrm{Z}(1) = \sum \mathrm{MQ}(\mathrm{M}, \mathrm{r}) \quad (3)$$

$$\mathrm{Z}(2) = \sum \mathrm{M}^2\mathrm{Q}(\mathrm{M}, \mathrm{r}) \quad (4)$$

Then to calculate lacunarity with box size r:

$$\mathrm{Lacunarity}(\mathrm{r}) = \frac{\mathrm{Z}(2) - (\mathrm{Z}(1))^2}{(\mathrm{Z}(1))^2} \quad (5)$$

The boxes-size used in this study were 2^1, 2^2, ..., 2^9.

2-4-3- Invariant Moment

Invariant moments are seven unchanging spatial moments in the continuous image domain for translation, rotation, and scale changes. The invariant moment was developed by Hu [32] and Wu et al.[33]. If the image function is expressed in $f(x,y)$, then the image with a height N and width M, then for geometric moments of order (p + q)th can be expressed

$$m_{pq} = \sum_{y=1}^{N}\sum_{x=1}^{M} x^p\, y^q\, f(x,y) \;\; p,q = 0,1,2.. \quad (6)$$

$$\bar{x} = m_{10}/m_{00} \;\; , \bar{y} = m_{01}/m_{00} \quad (7)$$

$$\mu_{pq} = \sum_{y=1}^{N}\sum_{x=1}^{M} (x-\bar{x})(y-\bar{y}) f(x,y) \quad (8)$$

The normalized central moment is defined as

$$\eta_{pq} = \mu_{pq}/\mu_{00}^{\frac{p+q}{2}+1} \quad (9)$$

By using the 2nd and 3rd normalized central moments orders to build 7 invariant moments M_1-M_7.

2-5- Feature Selection

Information Gain is a feature selection that is used to determine how significant the effect of a feature [34]. In this research, Information Gain will be used as a filter to filter out what features will be used in classification. These features are fractal dimension, lacunarity with a number of box sizes, and invariant moment consisting of 7 features. The value of the Information Gain is calculated using entropy. The calculation formula is as follows[34].

$$Entropy(D) = -\sum_{i}^{c} p_i \log_2 p_i \quad (10)$$

$$Gain(D,A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} Entropy(D_v) \quad (11)$$

where
c: The number of values present in the target attribute
p_i: the number of samples in class i
A: Attribute
V: a possible value for feature A
Values (A): the set of possible values for attribute A
|Sv|: the number of samples for the value v
|S|: the total number of data samples
Entropy (Sv): entropy for samples that have value v

2-6- Classification

AdaBoost or Adaptive Boosting is a boosting algorithm introduced by Freund and Schapire [35]. The boosting algorithm is an ensemble learning which uses a combination of classifiers to get a better-combined classification model. AdaBoost has two variants in its development, namely AdaBoost.M1 and AdaBoost.M2. The difference between the two variants is in the handling of errors. Adaboost.M1 uses a weighted classification error while Adaboost.M2 uses a weighted pseudo-loss. In terms of classification Adaboost.M1 is better used in binary classification while Adaboost.M2 is used in multiclass classification [35], [36]. In this study, using Adaboost.M1, because the case raised is a binary classifier.

The steps in the Adaboost.M1 algorithm are as follows:

a. Input is a dataset $D = \{(x_1, y_1), \ldots, (x_m, y_m)\}$; label $y_i \in Y = \{1, \ldots, k\}$. Basic learning algorithm (weak leaner) and the number of iterations T.

b. Initialize the weights with $D_{(i)} = \frac{1}{m}$ for $i = 1, \ldots, m$

c. Iterate for $t = 1, \ldots, T$

1) Train the weak learner h_t from D using the weight distribution d_t.

$$h_t = L(D, D_t) \qquad (12)$$

2) Calculate the error from h_t

$$\varepsilon_t = Pr_{i-D_i}[h_t(x_i \neq y_i)] \qquad (13)$$

$$\varepsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i) \qquad (14)$$

if $\varepsilon_t \geq {}^1/_2$, then set it $T = t - 1$, cancel the loop.

3) Determines the weight of h_t

$$\alpha_t = \ln(\frac{1-\varepsilon_t}{\varepsilon_t}) \qquad (15)$$

4) Update weights

$$D_{t+1}(i) \qquad (16)$$
$$= \frac{D_t(i)}{Z_t} x \begin{cases} \exp(-\alpha_t) & if\ h_t(x_i) = y_i \\ \exp(\alpha_t) & if\ h_t(x_i) \neq y_i \end{cases}$$

Where $Z_t = \sum_i D_t(i)$ is the selected normalization constant so that $D_{t+1}(i)$ will be the function of distribution.

d. Final classifier output

$$H_{fin(x)} = arg\max_{y\epsilon Y} \sum_{i:h_t(x)=y} \alpha_t(i) \qquad (17)$$

The final hypothesis $H_{fin(x)}$ is the weighted linear threshold of the weak hypothesis. That is, if given an instance of x, then $H_{fin(x)}$ produces an output on label y which maximizes the total weight of the weak hypothesis predicting that label [37].

2-7- Performance Analysis

Performance testing is carried out after the classification process is complete. The features that have been selected with information gain will be used as input in the classification. The test used is the k-fold cross-validation method with a value of k=5. Performance analysis was performed using several parameters, namely sensitivity, accuracy, specificity, likelihood ratio positive, and AUC (Area Under Curves). The calculation of the performance parameters is shown in equation (18-23).

$$\text{Sensitivity (SN)} = \text{TP}_{\text{rate}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (18)$$

$$\text{Specificity(SP)} = \text{TN}_{\text{rate}} = \frac{\text{TN}}{\text{TN} + \text{FP}} \qquad (19)$$

$$\text{Accuracy(AC)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (20)$$

$$\text{AUC (AU)} = \frac{1 + \text{TP}_{\text{rate}} - (1 - \text{TN}_{\text{rate}})}{2} [38] \qquad (21)$$

$$\text{Likelihood ratio} + \text{(LR+)} = \frac{\text{Sensitivity}}{1 - \text{Specificity}} \qquad (22)$$

$$\text{Likelihood ratio} - \text{(LR−)} = \frac{1 - \text{Sensitivity}}{\text{Specificity}} \qquad (23)$$

Where the TP means positive RH detected as positive RH. The FN is positive RH which is detected as negative RH. The FP is negative RH detected as positive RH, while TN is negative RH detected as negative RH. The LR (+) is the ratio between the probability of a positive test in individuals with the disease and the probability of a positive test in individuals without the disease. The LR (-) is the ratio between the probability of a negative test result in a diseased individual with the probability of a negative test result in an individual without the disease[39].

3- Result and Discussion

Referring to Figure 1, the results at the preprocessing stage for the retinal image conversion process into 3 image channels, namely green, blue, and read channel. The results at this stage can be shown in Figure 2. The next process is to remove the optical disc, with the results as shown in Figure 3. The final process of the preprocessing stage is contrast stretching, the result is as shown in Figure 4.

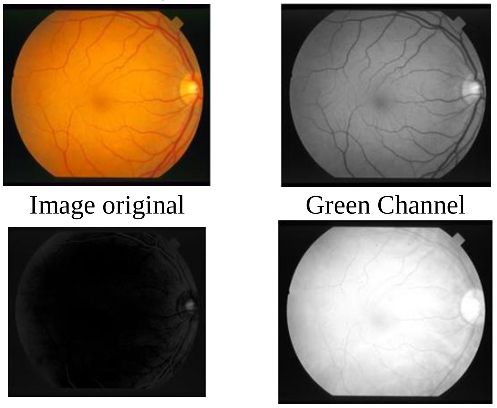


Image original Green Channel

Blue channel Read channel

Fig. 2: Retina Image & Gray Image

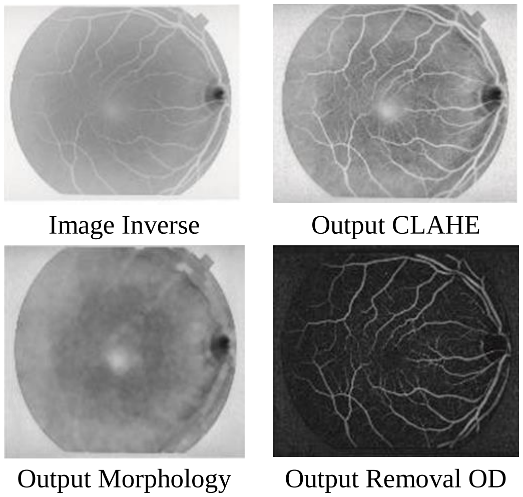


Image Inverse
Output CLAHE
Output Morphology
Output Removal OD

Fig. 3: Optic Disc Removal

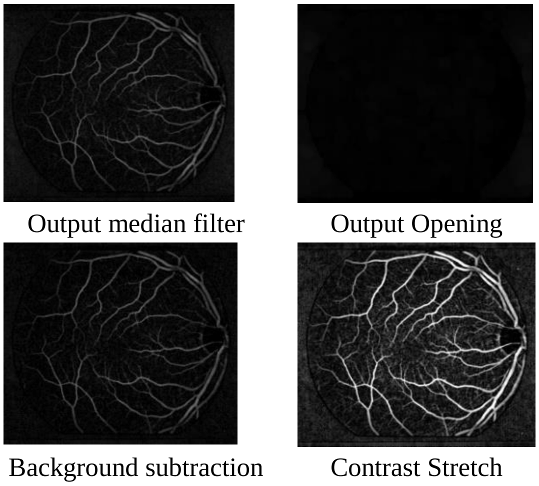

Output median filter
Output Opening
Background subtraction
Contrast Stretch

Fig. 4: Output contrast stretch

The segmentation process was carried out on 50 retinal images. Image segmentation is performed using the SOM clustering algorithm. The SOM algorithm uses neighbor parameter 3, 200 iterations, and the number of cluster centers tested is 2 to 10. The results of the image segmentation process using the number of clusters 5 can be seen in Figure 5. Furthermore, for other retinal image segmentation results for the number of clusters in the SOM algorithm between 2 to 7 can be shown in Figure 6.

The next stage is the feature extraction process, where the output at this stage can be shown in Table 1, by taking the number of clusters as an example 5. Table 1 shows the average value for each feature, both hypertensive or normal retinopathy. Table 1 also shows the results of statistical testing with a significance level of 95%. The next step is feature selection, using the information gain algorithm. The results of this process can be shown in Table 2.

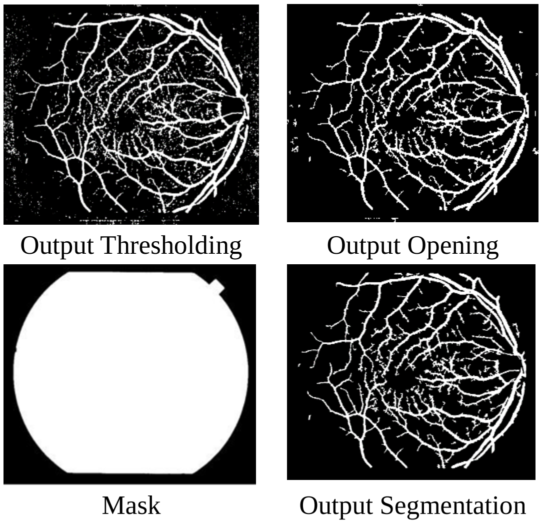

Output Thresholding
Output Opening
Mask
Output Segmentation

Fig. 5: Output segmentation process

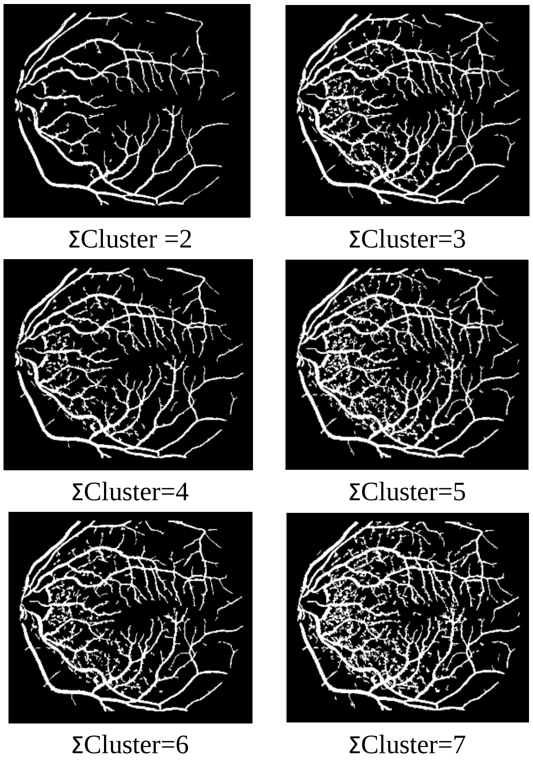

ΣCluster =2
ΣCluster=3
ΣCluster=4
ΣCluster=5
ΣCluster=6
ΣCluster=7

Fig. 6: The results of segmentation are based on the number of clusters

The next test result is the result of the classification process. The classification results are measured by the performance parameters of sensitivity, specificity, likelihood ratio positive, and area under the curve, shown in Table 3. Table 3 shows the performance for testing with clustering variables, and the best number of features.

Table 1 : Average feature extraction yield

No	Feature	RH	Normal	p-value
1	FD	1.602208	1.575276	0.001122
2	2^1	5.012032	6.029760	0.001182
3	2^2	4.301156	5.183120	0.001376
4	2^3	3.356580	4.024624	0.002155
5	2^4	2.432772	2.831768	0.007441
6	2^5	1.822812	2.018344	0.063592
7	2^6	1.433240	1.529700	0.220680
8	2^7	1.114265	1.172235	0.229945
9	2^8	0.861410	0.890279	0.058108
10	2^9	0.499498	0.499787	0.591386
11	M_1	0.582954	0.668611	0.018811
12	M_2	0.012519	0.011148	0.531156
13	M_3	0.004236	0.011822	0.019364
14	M_4	0.005215	0.010684	0.192822
15	M_5	0.000015	0.000048	0.667529
16	M_6	0.000232	0.000031	0.328706
17	M_7	0.000002	-0.000047	0.614520

Table 2 : Information Gain Results

Rank	2 Clusters		5 Clusters	
	score	Feature	Score	Feature
1	0.395816	FD	0.270252	FD
2	0.360657	2^1	0.222020	2^1
3	0.327324	2^2	0.208735	2^3
4	0.327324	2^3	0.182119	M_1
5	0.236453	2^4	0.182119	2^4
6	0.156513	M_1	0.173600	2^2
7	0.156513	2^5	0.124511	M_3
8	0.124511	2^7	0.106740	M_6
9	0.124511	M_3	0.096311	2^5
10	0.117188	2^8	0.087804	2^7
11	0.106740	2^6	0.087736	2^8
12	0.082296	M_5	0.087736	M_4
13	0.076591	M_2	0.085438	M_5
14	0.068648	2^9	0.085024	M_7
15	0.052821	M_6	0.069342	2^6
16	0.041203	M_4	0.051262	2^9
17	0.025695	M_7	0.041203	M_2

The CAD-RH system model, which is a hybrid SOM with AdaBoost, has the best performance when the number of SOM clusters is 2. The resulting performance is able to have an AUC value of 86%, or is included in the good category [40]. The resulting performance requires a relatively large number of features, namely 11 features. In addition to the number of clusters 2, the performance of CAD-RH, in the number of clusters of 5, is also able to provide performance with AUC values> 80%. The advantage of the number of clusters 5 is that it only requires 3 features, namely the fractal dimensions, the lacunarity with the size box are 2^1 and 2^3. The weakness of the number of clusters 5 is that the specificity value has a large difference compared to the number of clusters 2. Referring to the statistical test using the t-test method with 95% significance shown in Table 1, it can also be believed that 11 features and 3 features have a significant difference between positive and negative of hypertension retinopathy. This shows that the ranking generated by the information gain has similarities with the results of the t-test.

Table 3 : Classification Results in each Cluster

The number of		LR+	SN	SP	AU
cluster	feature				
2	11	7.00	84	88	86
3	4	2.18	96	56	76
4	2	2.11	76	64	70
5	3	4.20	84	80	82
6	17	2.86	80	72	76
7	5	3.17	76	76	76
8	17	2.38	76	68	72
9	3	1.25	80	36	58
10	17	1.08	52	52	52

The CAD-RH system in the number of clusters 2 with an AUC performance of 86% shows that, when the system is used to detect 100 patients, the system is able to detect as many as 86 patients with true positive hypertensive retinopathy. In the number of clusters, the AUC value was 82%, which means it was able to detect correctly for 82 patients. The performance on the number of clusters 2 and 5 has the same sensitivity performance parameters. Sensitivity is the ability of the CAD-RH system to identify positive patients with hypertensive retinopathy, identified by the CAD-RH system with positive results of hypertensive retinopathy. This is when used for initial screening, the sensitivity parameter becomes vital. In this hybrid model, the highest sensitivity occurs in the number of clusters 3, namely 96%, however, the specificity value is very low. So the ability of the CAD-RH system is low when identifying negative patients, the system is identified as negative.

When the number of clusters 5 shows that the invariant moment feature does not provide a significant additional performance. The test also shows that the tortuosity vascular pattern features can be extracted properly. When using fractal analysis, namely the fractal dimension and lacunarity. This is also supported by the results of feature selection with the information gain method, where the invariant moment, both M_1-M_7, has a relatively low entropy value compared to the fractal and lacunarity dimensions. This condition also proves the relationship between hypertensive retinopathy with fractal and lacunarity dimensions[12], [13].

Hybrid SOM with AdaBoost on the CAD-RH system shows relatively good capabilities with the resulting performance parameters, both when the number of clusters is 2 and the number of clusters 5. AUC's performance is in the range of 80%-90%, so it is categorized as good [40]. Referring to the research by McGee [39], that is, one of the performance parameters in the diagnostic system is the positive (LR +) and negative (LR-) likelihood ratio. This parameter is not limited to a scale of 0-100. Referring to these parameters, the value of LR +, for the number of clusters 2 with 11 features shows LR + = 7 and LR- = 0.182, while for the number of clusters 5 with 3 features LR + = 4.2 and LR- = 0.2, as shown in Table 3. The value of LR + will be better the higher the value, while LR- will be smaller the value. This also shows the performance in the number of clusters 2 is better.

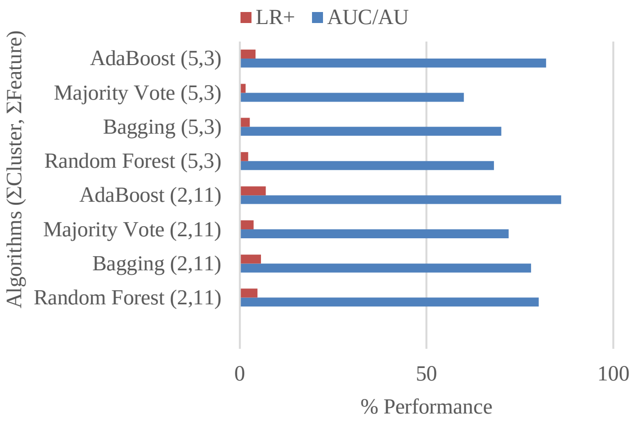


Fig. 7 : Comparison of Algorithms

The use of AdaBoost algorithm in the CAD-RH system shows more capabilities when compared to other ensemble algorithms. The comparison of AdaBoost with other algorithms can be shown in Figure 7. The ability that approaches the AdaBoost algorithm is Random Forest which is able to have AUC 80% when the number of clusters is 2. When compared to AdaBoost When the number of clusters is 5, Random Forest is superior when referring to the LR + parameter, whereas Lower AUC Random Forest. This difference was caused by the better random forest specificity value, but lower sensitivity. This means that there is a Boost with a number of clusters of 5, still better than Random Forest. This is also supported in research conducted by Stella et al.[42] and also Prastyo et al.[43], but in a different case. In this study, a number of algorithms were compared, including random forest, AdaBoost, and support vector machine (SVM). The results showed that there was an AdaBoost better than random forest and SVM.

4- Conclusions

The CAD-RH system model with SOM and AdaBoost has good performance when using the number of clusters 2 and the number of clusters 5. The number of clusters is able to provide good performance. The AUC value for the number of clusters 2 was 86% while for the number of clusters 5 the value was 82%. If we refer to the number of features, it achieved AUC more than 80% needed 3 feature when the number of clusters 5. While for the number of clusters 2 requires feature 11. Referring to the resulting performance, the Hybrid SOM and AdaBoost models can be an alternative in the initial diagnosis. hypertensive retinopathy.

Acknowledgments

We thank the Universitas Sebelas Maret for providing MRG research grants with contract number Nomor : 260/UN27.22/HK.07.00/2021. We also say thank you to the number of parties who have helped us complete the research that we did.

References

[1] W. Wiharto and E. Suryani, "The Review of Computer Aided Diagnostic Hypertensive Retinopathy Based on The Retinal Image Processing," in *The 2nd Sriwijaya international Conference on Science, Engineering, and Technology [SICEST]*, Palembang, Indonesia, 2018, vol. 690, pp. 1–9. doi: 10.1088/1757-899X/620/1/012099.

[2] F. Garcia-Lamont, J. Cervantes, A. López, and L. Rodriguez, "Segmentation of images by color features: A survey," *Neurocomputing*, vol. 292, pp. 1–27, May 2018, doi: 10.1016/j.neucom.2018.01.091.

[3] W. Wiharto and E. Suryani, "The Analysis Effect of Cluster Numbers On Fuzzy C-Means Algorithm for Blood Vessel Segmentation of Retinal Fundus Image," in *IEEE The 2nd International Conference on Information and Communications Technology*, Yogyakarta, Indonesia, 2019, pp. 1–5. doi: 10.1109/ICOIACT46704.2019.8938583.

[4] W. Wiharto and E. Suryani, "The Segmentation Analysis of Retinal Image Based on K-means Algorithm for Computer-Aided Diagnosis of Hypertensive Retinopathy," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 8, no. 2, pp. 419–426, 2020, doi: 10.11591/ijeei.v8i2.1287.

[5] W. Wiharto, E. Suryani, and M. Susilo, "The Hybrid Method of SOM Artificial Neural Network and Median Thresholding for Segmentation of Blood Vessels in the Retina Image Fundus," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 19, no. 4, pp. 323–331, 2019.

[6] F. Shafiei and S. Fekri-Ershad, “Detection of Lung Cancer Tumor in CT Scan Images Using Novel Combination of Super Pixel and Active Contour Algorithms,” *Traitement du Signal*, vol. 37, no. 6, pp. 1029–1035, Dec. 2020, doi: 10.18280/ts.370615.

[7] C. Lupascu and D. Tegolo, “Automatic unsupervised segmentation of retinal vessels using self-organizing maps and k-means clustering,” in *Computational Intelligence Methods for …*, Berlin, Heidelberg, 2011, vol. 6685, pp. 263–274. doi: 10.1007/978-3-642-21946-7_21.

[8] C. Budayan, I. Dikmen, and M. T. Birgonul, “Comparing the performance of traditional cluster analysis, self-organizing maps and fuzzy C-means method for strategic grouping,” *Expert Systems with Applications*, vol. 36, no. 9, pp. 11772–11781, 2009, doi: 10.1016/j.eswa.2009.04.022.

[9] S. Arumugadevi and V. Seenivasagam, “Comparison of Clustering Methods for Segmenting Color Images,” *Indian Journal of Science and Technology*, vol. 8, no. 7, pp. 670–677, 2015, doi: 10.17485/ijst/2015/v8i7/62862.

[10] O. A. Abbas, “Comparisons Between Data Clustering Algorithms,” *The International Arab Journal of Information Technology*, vol. 5, no. 3, pp. 320–325, 2008.

[11] K. K. Jassar, “Comparative Study and Performance Analysis of Clustering Algorithms,” *IJCA Proceedings on International Conference on ICT for Healthcare*, vol. ICTHC 2015, no. 1, pp. 1–6, 2016.

[12] P. Zhu *et al.*, “The relationship of retinal vessel diameters and fractal dimensions with blood pressure and cardiovascular risk factors,” *PLoS ONE*, vol. 9, no. 9, pp. 1–10, 2014, doi: 10.1371/journal.pone.0106551.

[13] N. Popovic, M. Radunovic, J. Badnjar, and T. Popovic, “Fractal dimension and lacunarity analysis of retinal microvascular morphology in hypertension and diabetes,” *Microvascular Research*, vol. 118, no. 2018, pp. 36–43, 2018, doi: 10.1016/j.mvr.2018.02.006.

[14] H. A. Crystal *et al.*, “Association of the Fractal Dimension of Retinal Arteries and Veins with Quantitative Brain MRI Measures in HIV-Infected and Uninfected Women,” *PLoS ONE*, vol. 11, no. 5, pp. 1–11, 2016, doi: 10.1371/journal.pone.0154858.

[15] E. V. L. Costa and R. A. Nogueira, “Fractal, multifractal and lacunarity analysis applied in retinal regionsof diabetic patients with and without non-proliferative diabetic retinopathy,” *Fractal Geometry and Nonlinear Anal in Med and Biol*, vol. 1, no. 3, pp. 112–119, 2016, doi: 10.15761/FGNAMB.1000118.

[16] W. Wiharto, E. Suryani, and M. Yahya Kipti, “Assessment of Early Hypertensive Retinopathy using Fractal Analysis of Retinal Fundus Image,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 1, pp. 445-454, 2018, doi: 10.12928/telkomnika.v16i1.6188

[17] M. F. Syahputra, I. Aulia, R. F. Rahmat, and others, “Hypertensive retinopathy identification from retinal fundus image using probabilistic neural network,” in *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*, 2017, pp. 1–6.

[18] K. Narasimhan, V. C. Neha, and K. Vijayarekha, “Hypertensive retinopathy diagnosis from fundus images by estimation of AVR,” *Procedia Engineering*, vol. 38, no. 2012, pp. 980–993, 2012, doi: 10.1016/j.proeng.2012.06.124.

[19] N. Hutson, A. Karan, J. A. Adkinson, P. Sidiropoulos, I. Vlachos, and L. Iasemidis, “Classification of Ocular Disorders Based on Fractal and Invariant Moment Analysis of Retinal Fundus Images,” in *2016 32nd Southern Biomedical Engineering Conference (SBEC)*, Shreveport, LA, USA, Mar. 2016, pp. 57–58. doi: 10.1109/SBEC.2016.21.

[20] U. G. Abbasi and U. M. Akram, “Classification of blood vessels as arteries and veins for diagnosis of hypertensive retinopathy,” in *2014 10th International Computer Engineering Conference: Today Information Society What’s Next?, ICENCO 2014*, Giza, Egypt, 2014, pp. 5–9. doi: 10.1109/ICENCO.2014.7050423.

[21] B. K. Triwijoyo, W. Budiharto, and E. Abdurachman, “The Classification of Hypertensive Retinopathy using Convolutional Neural Network,” *Procedia Computer Science*, vol. 116, pp. 166–173, 2017, doi: 10.1016/j.procs.2017.10.066.

[22] X. Miao and J. S. Heaton, “A comparison of random forest and Adaboost tree in ecosystem classification in east Mojave Desert,” in *2010 18th International Conference on Geoinformatics*, Beijing, China, Jun. 2010, pp. 1–6. doi: 10.1109/GEOINFORMATICS.2010.5567504.

[23] C. A. Lupascu, D. Tegolo, and E. Trucco, “FABC: Retinal Vessel Segmentation Using AdaBoost,” *IEEE Trans. Inform. Technol. Biomed.*, vol. 14, no. 5, pp. 1267–1274, Sep. 2010, doi: 10.1109/TITB.2010.2052282.

[24] N. Dey, A. B. Roy, M. Pal, and A. Das, “FCM Based Blood Vessel Segmentation Method for Retinal Images,” *International Journal of Computer Science and Network (IJCSN)*, vol. 1, no. 3, pp. 1–5, 2012.

[25] G. B. Kande, T. S. Savithri, and P. Subbaiah, “Segmentation of vessels in fundus images using spatially weighted fuzzy c-means clustering algorithm,” *International journal of computer science and network security*, vol. 7, no. 12, pp. 102–109, 2007.

[26] R. A. Aras, T. Lestari, H. Adi Nugroho, and I. Ardiyanto, “Segmentation of retinal blood vessels for detection of diabetic retinopathy: A review,” *Communications in Science and Technology*, vol. 1, no. 2016, pp. 33–41, 2016, doi: 10.21924/cst.1.1.2016.13.

[27] Ș. Talu, C. Vlăduțiu, L. A. Popescu, C. A. Lupașcu, Ș. C. Vesa, and S. D. Țălu, “Fractal and lacunarity analysis of human retinal vessel arborisation in normal and amblyopic eyes,” *Human and Veterinary Medicine*, vol. 5, no. 2, pp. 45–51, 2013.

[28] T. Acharya and K. R. Ray, *Image Processing: Principles and Applications*. USA: John Willey & Sons, 2005.

[29] B. B. Mandelbrot, *The fractal geometry of nature*, vol. 173. WH freeman New York, 1983.

[30] A. R. Backes and O. M. Bruno, “A new approach to estimate fractal dimension of texture images,” in *In: Elmoataz A., Lezoray O., Nouboud F., Mammass D. (eds) Image and Signal Processing. ICISP 2008. Lecture Notes in Computer Science*, Berlin, Heidelberg, 2008, vol. 5099, pp. 136–143. doi: 10.1007/978-3-540-69905-7_16.

[31] C. Allain and M. Cloitre, “Characterizing the lacunarity of random and deterministic fractal sets,” *Physical Review A*, vol. 44, no. 6, pp. 3552–3558, 1991, doi: 10.1103/PhysRevA.44.3552.

[32] M.-K. Hu, “Visual pattern recognition by moment invariants,” *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.

[33] Z. Wu *et al.*, “Application of image retrieval based on convolutional neural networks and Hu invariant moment algorithm in computer telecommunications,” *Computer Communications*, vol. 150, pp. 729–738, Jan. 2020, doi: 10.1016/j.comcom.2019.11.053.

[34] S. Lefkovits and L. Lefkovits, “Gabor Feature Selection Based on Information Gain,” *Procedia Engineering*, vol. 181, no. 2017, pp. 892–898, 2017, doi: 10.1016/j.proeng.2017.02.482.

[35] Y. Freund and R. E. Schapire, “Experiments with a New Boosting Algorithm,” in *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, The University of Virginia, 1996, pp. 148–156.

[36] G. Eibl and K. Pfeiffer, “Multiclass Boosting forWeak Classifiers,” *Journal of Machine Learning Research*, vol. 6, no. 7, pp. 189–210, 2005.

[37] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/jcss.1997.1504.

[38] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, “SMOTE-RSB *: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory,” *Knowledge and Information Systems*, vol. 33, no. 2, pp. 245–265, 2012, doi: 10.1007/s10115-011-0465-6.

[39] S. McGee, “Simplifying likelihood ratios,” *J Gen Intern Med*, vol. 17, no. 8, pp. 647–650, Aug. 2002, doi: 10.1046/j.1525-1497.2002.10750.x.

[40] F. Gorunescu, *Data Mining: Concepts, Models and Techniques*. Berlin, Heidelberg: Springer, 2011.

[41] M. Stella and S. Kumar, “Prediction and Comparison using AdaBoost and ML Algorithms with Autistic Children Dataset,” *IJERT*, vol. 9, no. 7, pp. 133–136, Jul. 2020, doi: 10.17577/IJERTV9IS070091.

[42] P. H. Prastyo, I. G. Paramartha, M. S. M. Pakpahan, and I. Ardiyanto, “Predicting Breast Cancer: A Comparative Analysis of Machine Learning Algorithms,” *PROC. INTERNAT. CONF. SCI. ENGIN.*, vol. 3, no. 2020, pp. 455–459, 2020.

Wiharto is an Associate professor of Computer Science at Department of Informatics, Sebelas Maret University, Surakarta, Indonesia. He received his Ph.D. degree in Biomedical Engineering (Medical Informatics) from Gadjah Mada University, Indonesia in 2017. He is conducting research activities in the areas of Artificial Intelligence, Computational Intelligence, Medical Imaging, Clinical Decision Support System and Data Mining.

Esti Suryani received obtained a Bachelor of Science (B.S.) from Gadjah Mada University, Yogyakarta, Indonesia, 2002 and Master’s Degree in Computer Science (M.Cs.) from Gadjah Mada University, Yogyakarta, Indonesia, 2006. He is presently working as an Assistant professor in the Department of Informatics, Faculty of mathematics and natural sciences, Sebelas Maret University, Surakarta, Indonesia. His experience and areas of interest focus on image processing, fuzzy logic, and Data security.

Murdoko Susilo received obtained a Bachelor of Science (B.S.) from Sebelas Maret University, Surakarta, Indonesia, 2020. The area of research being carried out is the medical image processing, data mining, artificial intelligence, and clinical decision support system

Utilizing Gated Recurrent Units to Retain Long Term Dependencies with Recurrent Neural Network in Text Classification

Nidhi Chandra*
Research Scholar, Amity Institute of Information Technology, Amity University Uttar Pradesh, India
Srivastavanidhi8@gmail.com
Dr. Laxmi Ahuja
Dy. Director, Amity Institute of Information Technology, Amity University Uttar Pradesh, India
lahuja@amity.edu
Dr. Sunil K. Khatri
Director, Amity University Tashkent, Uzbekistan.
skkhatri@amity.edu
Dr. Himanshu Monga
Professor, Jawaharlal Lal Nehru Government Engineering College, Sundernagar, Mandi (H.P).
himanshumonga@gmail.com

Received: 11/Jul/2020 Revised: 21/Apr/2021 Accepted: 12/May/2021

Abstract

The classification of text is one of the key areas of research for natural language processing. Most of the organizations get customer reviews and feedbacks for their products for which they want quick reviews to action on them. Manual reviews would take a lot of time and effort and may impact their product sales, so to make it quick these organizations have asked their IT to leverage machine learning algorithms to process such text on a real-time basis. Gated recurrent units (GRUs) algorithms which is an extension of the Recurrent Neural Network and referred to as gating mechanism in the network helps provides such mechanism. Recurrent Neural Networks (RNN) has demonstrated to be the main alternative to deal with sequence classification and have demonstrated satisfactory to keep up the information from past outcomes and influence those outcomes for performance adjustment. The GRU model helps in rectifying gradient problems which can help benefit multiple use cases by making this model learn long-term dependencies in text data structures. A few of the use cases that follow are – sentiment analysis for NLP. GRU with RNN is being used as it would need to retain long-term dependencies. This paper presents a text classification technique using a sequential word embedding processed using gated recurrent unit sigmoid function in a Recurrent neural network. This paper focuses on classifying text using the Gated Recurrent Units method that makes use of the framework for embedding fixed size, matrix text. It helps specifically inform the network of long-term dependencies. We leveraged the GRU model on the movie review dataset with a classification accuracy of 87%.

Keywords: Gated Recurrent Units; Recurrent Neural Network; Word Embedding; Deep Learning; LSTM.

1- Introduction

It has been more than a decade that we are communicating with machines through handwritten codes or programs. It has been always a human dream that machines should understand their language or in other words, they should be able to speak to machines and machines should respond in the same language. Natural language processing helped to make this dream come true. Natural language processing is analyzing and making digital sense to the natural language spoken by humans across the geographies or it is also referred to as machine processing of human languages to channelize interaction between human and machine. There are multiple applications to NLP such as IVR systems integrated with chatbots, standalone chatbots, etc. Where an individual can connect through his phone and speak his query, which is then translated into machine understandable form and processed to address end-user query. There are live robots these days which not only understand language types but also respond in similar language.

NLP uses numerical and statistical techniques to convert textual data into numerical data which machines could understand and maps this data to machine learning and deep learning algorithms to help to bridge the gap of communication between humans and machines.

The text has to be processed in multiple stages or phases before making it machine-readable or understandable also

* Corresponding Author

known as text processing or filtering and helps to serve multiple purposes. This process is corpora dependent and requires text preparation to be performed to make it enable to input into an appropriate machine or deep learning algorithms.

Because of the dependency of machine learning and deep learning algorithms on numerical data to give the best results, word embedding [1][14] plays a critical role to transform preprocessed corpora into numerical types. Word embedding leverages real-value vector representations of words which supports models in predicting and understanding words. The two key algorithms to be used for this purpose are – Word2Vec and Glove [17] [29][40].

Another technique that follows the same approach as machine learning is deep learning which leverages artificial neural networks as computing models. The ANN [27][22] technique is inspired by the network of neurons in the human brain and how they store information in a form of layers. It also tries to depict how the information is retrieved from these neural nets. In computer science implementations these neural nets were implemented as connected nodes forming a network similar to that inside the human brain. These nodes are responsible to learn and store information like text, real-life objects Etc. The neural nets are a collection of layers whose numbers can range from three to hundreds. They are further classified as shallow and deep neural networks depending on the number of layers in them. Shallow networks are confined to three to four layers while deep networks have more than four layers. Because of a greater number of layer processing deep learning models are preferred over shallow networks in complex tasks like facial recognition, text translation, etc.

2- Literature Survey

Lately, there is a functioning pattern towards utilizing different AI strategies for taking care of issues identified with Natural Language Processing (NLP). One of these issues is the programmed recognition of emotion. The investigation of sentiment and emotions has an elaborative history. Sentiment analysis is contextual mining of text which distinguishes and extricates emotional data in the source material and helping a business to comprehend the social feeling of their image, item, or administration while observing on the web discussions.

No such extensive survey exists which should discuss various approaches which researchers are applying to identify the shortfall of an exhaustive report to investigate the collection of different patterns.

This comprehensive survey [12] results depicts the entire, agreed upon, and planned review of views or judgment and emoticon analysis for classification of methods to have comparative analysis for better comprehensions.

Socher et al. [34] has presented a hierarchical structure that centers on the perspective explicit investigation of emotions. To extract labels at the phrase level, they created novel d-dimensional vector portrayals for terms**,** built up a profound learning framework including managing highlight**,** portrayals of sentence parses, which adds to the assurance of a target task Comparison is made of multi-vector RNN and recursive neural tensor system nearby vanilla RNN for this. Using their collaborative multi-aspect feeling layout, these are appended to perspective and feeling labels. They differentiated their sentiment pair recognition model for single and joint viewpoint and differentiated it against multiclass SVMs and Naive Bayes classifiers.

Junyoung et. al. [19] proposed a novel architecture for deep-stacked RNNs. Their study Suggested
RNN, gated-feedback RNN (GF-RNN), expands the existing method of stacking multiple recurring layers by enabling and regulating signals flowing from the upper recurrent layers to lower layers using a global gating unit for each pair of layers. Experiments focused on challenging sequence modeling tasks of character-level language modeling.

Liu et al. [13] presented a hybrid method for bilingual text sentiment classification based on a deep learning approach that combined machine learning with deep learning to provide a stronger result in recognition of feelings.

Chen, Huimin, et al. [5] presented a model that first builds a hierarchical LSTM model to generate sentence and document representations. Afterward, user and product information is considered via attention over different semantic levels due to its ability to capture crucial semantic components. This paper proposes a hierarchical neural network that incorporates user and product information via word and sentence level attentions. With the user and product attention, our model can take account of the global user preference and product characteristics at both word-level and semantic levels.

The consolidated set of words are processed through the process of embedding layer wherein each token is classified as a variable-sized vector with actual meaning, which is many of times referred to as word embedding. There have been experimentations on few specific methods of word embedding initialization such as random Glove [11][36] and SSWE [11]. For the preparation one can improve the model base on the embedding of word learning through word function.

Johnson and Zhang [20] suggested a CNN variant called BoW-CNN which uses bag-of-word conversion in the convolution layer. They also presented a model, called

Seq-CNN, which preserves sequential word knowledge by concatenating the multiple word one-hot vectors.

Tang et al. [32] proposed a neural network to learn the representation of documents, considering the relation between sentences. Next, it learns from word embedding's the sentence representation with CNN or LSTM. A GRU is then used for the adaptive encoding of sentence semantics and their underlying relationships in text representations for the classification of sentiments.

Tang et al. [33] user representations applied, and product representations listed in the analysis. The hope is that such representations can capture essential global clues like individual user expectations and overall product quality which can provide better representations of text.

O Yildirim [28] presented a new model for deep bidirectional LSTM network-based wavelet sequences called DBLSTM-WS was proposed for classifying electrocardiogram (ECG) signals. The new wavelet-based layer is implemented to generate ECG signal sequences. The ECG signals were decomposed into frequency sub-bands at different scales in this layer.

These sub-bands are used as sequences for the input of LSTM networks. New network models that include unidirectional (ULSTM) and bidirectional (BLSTM) structures are designed for performance comparisons. Experimental studies have been performed for five different types of heartbeats obtained from the MIT-BIH arrhythmia database.

B. Athiwaratkun et. al. [4] propose several new malware classification architectures which include a long short-term memory (LSTM) language model and a gated recurrent unit (GRU) language model. He proposed an attention mechanism in addition to temporal max-pooling as an alternative way to construct the file representation from neural features. A new single-stage malware classifier based on a character-level convolutional neural network (CNN) is proposed in this study. Results show that the LSTM with temporal max pooling and logistic regression offers a 31.3% improvement in the true positive rate.

M Zulqarnain et. al. [25] proposed a unified structure to investigate the effects of word embedding and Gated Recurrent Unit (GRU) for text classification on two benchmark datasets included (Google snippets and TREC). GRU is a well-known type of recurrent neural network (RNN), which is the ability to compute sequential data over its recurrent architecture. First, words in posts are changed into vectors via the word embedding technique. Then, the words sequential in sentences are fed to GRU to extract the contextual semantics between words. The experimental results showed that the proposed GRU model can effectively learn the word usage in the context of texts provided training data

Luo, L et. al. [24] improve the performance of internet public sentiment analysis, a text sentiment analysis method combining Latent Dirichlet Allocation (LDA) text representation and convolutional neural network (CNN) is proposed. First, the review texts are collected from the network for preprocessing. Then, using the LDA topic model to train the latent semantic space representation (topic distribution) of the short text, and the short text feature vector representation based on the topic distribution is constructed. Finally, the CNN with the gated recurrent unit (GRU) is used as a classifier. According to the input feature matrix, the GRU-CNN strengthens the relationship between words and words, text and text, to achieve high accurate text classification.

Mostly NER is performed in English rather than Chinese, and the current impact of named entity recognition in Chinese is not very satisfactory due to the complex language characteristics of Chinese. S. Yan et al. suggest a new network framework called BERT-BGRU-MHA-CRF to address the above issues [30]. Experiments show that the model is capable of achieving an F1 value of 94.14.

To detect rumors, Zhou et al. used a combination of convolutional neural networks and gated recurrent unit networks [39]. Their model vectorizes rumor events, then uses CNN to automatically create rumor microblog features, and then uses GRU to mine temporal information between related microblogs under rumor events. The C-GRU model outperforms a sequence of classic models in the experiments.

Y. Pan et al. proposed the Bi-GRU (bidirectional GRU neural network) and attention mechanism model to analyze Chinese text sentiment to solve the problem of high complexity and low efficiency in Chinese LSTM-based Chinese text sentiment analysis [38]. To learn the text features more accurately, the model extracts deep features from the text and combine them with the meaning of the sentence. Multi-Head Self-Attention is implemented in the method, which eliminates dependency on external parameters, assigns weights to word vectors, highlights text attributes, and pays more attention to internal sentence dependencies. Experiments show that the model-based Chinese text sentiment analysis can achieve an accuracy of 87.1 percent.

H. Wang et al. suggested a hybrid learning-based emotion classification model [16]. To measure emotion scores in the entire data collection, the enhanced dictionary classification system is used, and data with high or low scores are explicitly marked; the rest of the method is based on emotion dictionary and Bi-GRU fusion model to calculate emotion score. The COAE2014 (Chinese opinion analysis and evaluation 2014) dataset microblog experiment with emoticons reveals that a single model is not suitable for several forms of diverse contexts, and it is difficult and inaccurate. The multi-model fusion method can effectively boost the single model's error preference and classification effect.

3- Review of Deep Learning

Deep learning helps us extend machine learning algorithms to improve results on text, image, and voice. The algorithms designed and implemented through deep learning draws similarities to stimuli and neurons of the human brain. One finds implementations of deep learning algorithms in electronic vision, computer-based translation, and recognitions.

It is a subfield for machine learning, as both approaches adopt the same primary principle — both machine learning and deep learning algorithms take input and use it to predict the output.

The objectives of Machine learning and deep learning algorithm(s) is to reduce differences between actual and predicted results after being trained on the training dataset. This helps to achieve higher accuracy by establishing the relationship between input and output. Machine learning models despite being self-sufficient need human inputs and feedbacks to confirm whether the prediction is correct or not by itself [23][35]. This, leading to the conclusion that deep learning models are autonomous, thereby making decisions to improve the effectiveness without human intervention. Even though neural networks were lost interest by the research fraternity in the late '90s being computationally very costly, but due to advancement in CPU technology in the past 10 years, several breakthroughs were made in this area and made progress in computer vision, speech, and NLP[10][15]. The key attributes considered for the revival of neural networks are the availability of high-speed computing resources like GPU and vast quantities of training data on the Internet [3].

Recurrent neural networks are used when a sequence of data shifts over time. It is like a conventional feed-forward network, the difference being that it has connections to the same layer of units. RNN can make use of the information in arbitrarily long sequences, but in practice, the standard RNN is limited to looking back only a few steps due to the vanishing gradient or exploding gradient problem [2].

The Long Short-Term Memory Network is a special RNN type capable of learning Dependencies in the long term [7]. LSTM and GRU solve the issue of vanishing gradient too. A slight variation of LSTM is the Gated Recurrent Unit [8].

To learn document representation, Tang et al. [32] suggested a neural network, with the consideration of Relationships in sentences. The sentence representation with CNN or LSTM is first learned from the word Embedded. Then a GRU is used to adaptively encode sentence semantics and their underlying semantics. Links in text representations for classification of emotion.

Chen et. al. [6] proposed to utilize a recurrent attention network to better capture the sentiment of complicated contexts. To achieve that, their proposed model uses a recurrent attention structure and learns a non-linear combination of the attention in GRUs.

Research	**Model Used**	**Dataset**	**Results**
[37] Turian, Joseph et. al.	Tree-LSTM	Sentences Involving Compositional Knowledge (SICK) dataset	Tree-LSTM used for sentiment classification of sentences sampled from movie reviews.
S. Yan et. al. [30]	BERT-BGRU-MHA-CRF	Chinese Language Scripts	Model can achieve F1 value of 94.14.
Zhou et al. [39]	CNN+GRU	Chinese blogs dataset	. The CNN-GRU model outperforms a sequence of classic models in the experiments .
Y. Pan et al. [38]	Bidirectional - GRU	Chinese scripts	Bidirectional - GRU can achieve an accuracy of 87.1 percent.
D. Tang et al. [11]	Conv GRNN and LSTM-GRNN	IMDB and Yelp Dataset	Gated recurrent neural network outperforms standard recurrent neural network in document

			modelling for sentiment classification
M. U. Salur et al. [26]	12 deep learning models and CNN, LSTM, Bi-LSTM, GRU	GSM operator Turkey dataset	CNN+ bi LSTM has higher accuracy i.e. 82.14% as compared to CNN.
] Johnson R et al. [20]	CNN on high-dimensional text data	IMDB dataset	CNN outperforms with, bag-of-n gram approach or word-vector.
Chen P et al. [6]	LSTM, Bi-LSTM, RNN, GRU	SemEval 2014, Tweet Dataset, Chinese news reports	Recurrent Attention on Memory (RAM) with 4 attentions and not trained data set, outperformed compared to rest multi attentions scheme

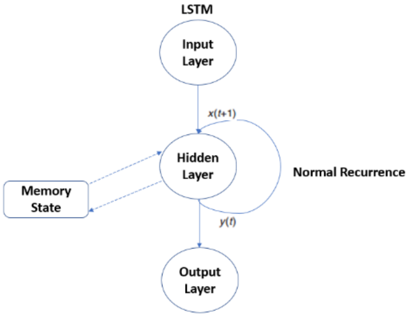
Table 1: Summary of Methods Related to Proposed Approach

The approach presented in this paper is based on [6][11] as depicted in Table 1 to build a GRU-based classifier for a binary polarized dataset [41].

4- Long Short-Term Memory Networks

In 1997, Sepp Hochreiter and Jürgen Schmidhuber first introduced LSTM networks and addressed the problem of retaining information for RNNs for longer periods [37].
RNNs have proven to be the only option to handle sequence classification issues and have proved acceptable to maintain the data from previous results and leverage those results for modification of outputs.
The saver here is an LSTM network which is an RNN architecture that helps to train models over lengthy sequences and to retain memories out of previous input time steps fed into the model.
This would help in handling the problem of gradient disappearance or explosion through the introduction of added gates, inputs, and forgetting gates, enabling better gradient regulations, helping to enable - Information to maintain and forget, thereby controlling the access to Information for the current cell's state helping in the preservation of "long-range dependencies"[2][31].

LSTM comprises memory cells to store information. It computes the input gate, forgets gate, and output gate to manage this memory. LSTM units can broadcast some critical features that inculcate early in the input sequence over a long distance thereby capturing critical long-distance dependencies [7] as depicted in Figure (1).



.

Figure 1. Long-Distance Dependencies

In LSTMs, the rules that govern the state-saved information (memory) are themselves trained neural networks — therein lies the magic. The network can be programmed to know what to recall, while the rest of the recurrent system will at the same time learn to predict the target.
Input influences the memory state (see Figure 1) and influences the layer output as well as a regular recurrent system. But the state of memory continues in all phases of the time series (your sentence or document). So, each input will impact both the memory state and the output of the hidden layer. The memory state mystery which it knows when to recall. Memory state wonders that when learning to reproduce the output by regular backpropagation, it knows what to remember.
Although complex, LSTM is very competitive in various tasks such as the recognition of handwriting, machine translation, and, of course, the analysis of sentiments.
They're typically slower than other standard ones, apart from the LSTM network complexity. Also, an RNN with better initialization and planning, and with less computational complexity, can produce results close to those of LSTM. Besides, where recent information is more important than older information, the LSTM model is often a better choice.

4-1- Gated Recurrent Units:

There are wide ranges of variants of LSTM that are in use today like the Gated recurrent unit (GRU) that creates an update gate through the consumption forget and input gates. It combines the state of the cell using a hidden state and modifies the generated output making target models less complex than regular ones.

GRU can control the flow of information without the need to use a memory unit like LSTM which is proved best to work with large datasets [32]. On the contrary GRU maps to smaller data sets. It is not mandatory though as efficiency would depend to some degree on the complexity of data and model.

4-2- GRU for Text Classifier

Sentiment/ emotion analysis is a common case of use when applying the techniques of natural language processing. Sentiment analysis aims to determine whether to interpret a given piece of text as conveying a 'positive' feeling or a 'negative' feeling.

"The movie was so awesome that I slept peacefully for hours."

To a human reader, it is painfully clear that the book review referred to here transmits a negative feeling. So, how do you build a model of machine learning to recognize sensations? As usual, the use of a supervised learning method requires a text corpus that contains many samples. Each piece of text in this corpus should have a label indicating whether a positive or negative emotion can be mapped to the text.

By looking at the example above, you can already see that such a function might be difficult for a machine learning model to solve. By using a simple tokenization or TFIDF approach, the classifier can quickly misinterpret words like 'wonderful' and 'peacefully' to convey a positive feeling.

To make matters worse, the text contains no word which can be translated explicitly as negative. This observation also brings in the need to link various sections of the text structure so that meaning can be taken from the sentence. The first sentence can be broken up into two parts, for example:

"The movie was so awesome"
"I slept peacefully for hours"

Looking at just the first part of the paragraph, you might infer it's a good remark. It is only when the second sentence is understood that the context of the sentence can be completely interpreted as expressing negative feelings. And there has to be long-term dependency here. And a simple RNN for mission is not good enough.

4-3- Structure of GRU

Gated Recurrent Unit (GRU) ascertains two gates that control data move through each hidden state, called update and reset gates. The existing input and a vector comprising of earlier hidden states should be taken as input. The value of three isolated [28] gates should be calculated by using the below steps –

1. The parameterized input and the hidden state vector need to be calculated for each gate through element-wise multiplication referred to as Hadamard multiplication between vector and weights of each gate.
2. The activation function needs to be applied to respective gates as element by element basis on the parameterized vector.

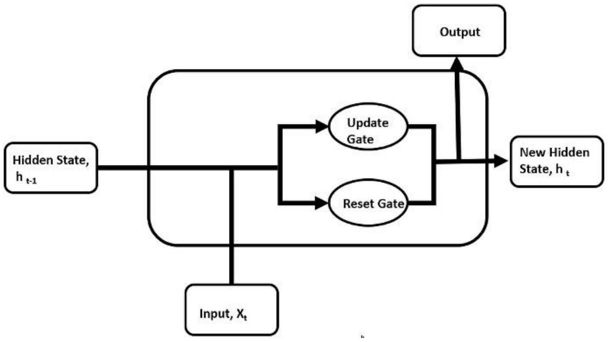


Figure 2: GRU Memory States

In contrast to the LSTM, GRU only has two gates Reset Gate and Update Gate as shown in figure (2). Each hidden state is determined in time-step t utilizing the accompanying conditions.

Reset Gate g_r as explained in Eq. (1) corresponds to the summation of input gate and forget gate in the LSTM network which helps us find the forgotten knowledge of the past. Here X_t represents the input vector and $h_{t-1}$is a hidden state at time t.

$$g_r = \sigma(W_{rx}X_t + W_{rh}h_{t-1} + b) \qquad (1)$$

Update gate g_u corresponds to the output gate of the LSTM recurrent unit which helps determine the previous knowledge which should be transit to the future. The sigmoid function is used as an activation function in the update gate as well as in the reset gate [32].

$$g_u = \sigma(W_{ux}X_t + W_{uh}h_{t-1} + b) \qquad (2)$$

As shown in Eq. 2 $\mathrm{h_t}$ represents the hidden state at time t and $\mathrm{h_{t-1}}$depicts the hidden state at time t-1

The new hidden state (proposed) state at time t, i.e. $\breve{h}_t$ is calculated as per the Eq. (3)

$$\breve{h}_t = tanh\,(W_{hx}\,X_t\; + W_{hh}\;\;.\,(\,g_r.\,h_{t-1}) + b) \qquad (3)$$

First, the Hadamard product of the Reset Gate and the previously hidden state vector as shown in Eq. 3 are calculated. Then this vector is parameterized and then added to the parameterized current input vector.

Hidden state at time t i.e. $\mathrm{h_t}$ is often ignored during the evaluation of GRU network and is often included into Reset Gate similar to Input Modulation gate which is a section of input gate and helps to incorporate non-linearity in input and makes it zero-mean as shown in figure (2). It also helps to decrease the effects of prior details on existing Information that are passed into the future.

$$h_t = (1 - g_r).\,h_{t-1} + g_u.\;\breve{h}_t \qquad (4)$$

In this Eq. (4), $\mathrm{g_r}$ as Reset gate, $\mathrm{g_u}$ as Update gate, and $\mathrm{h_t}$ as a hidden state at time t, $\mathrm{h_{t-1}}$= hidden state at time t-1, $\breve{\mathrm{h}}_\mathrm{t}$=New Hidden state at time t are taken to calculate the current memory gate.

GRU Gates incorporates initiations of the sigmoid. Like tanh activation, a sigmoid activation Instead of squishing values from -1 to 1 it squishes values from 0 to 1 as shown in figure (3). This is useful for updating or losing data as any number multiplying by 0 is 0, causing values to vanish or be "forgotten. "Any number multiplied by 1 is the same value so that the value remains the same or is "kept." Therefore, the system can realize which information isn't significant and which information ought to be overlooked or kept.

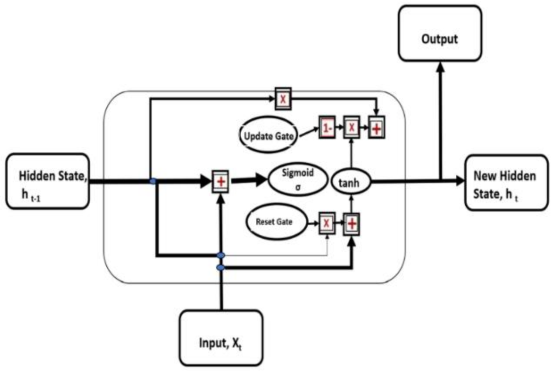


Figure 3: GRU Architecture and Functions

In the first place, we have the forget gate. This gate decides which information to throw away or to keep. Information from the previous hidden state is passed through the sigmoid function and information from the current input. Values range from 0 to 1, the closest to 0 means missing, and the closest to 1 means holding off.
We have the input gate to change cell status. Second, the previous hidden state and current input are passed into a sigmoid function. That determines what values are changed by converting the values between 0 and 1. 0 Does not mean important and 1 does mean importance. In the tanh method, you also transfer the hidden state and current input to squish values between -1 and 1 to help regulate the network. The tanh output is then multiplied by the sigmoid output. The sigmoid output will determine which information to retain from the tanh output is significant.
Gates are merely neural networks that regulate the flow of information through the sequence chain.

In the first place, we have the overlook entryway. This door chooses which data to discard or to keep. Data from the past shrouded state is gone through the sigmoid capacity and data from the current information. Qualities extend from 0 to 1, the nearest to 0 methods missing and the nearest to 1 method holding off.

5- Experimental Setup

The experiments were done on the Anaconda environment which is an open-source package manager, environment manager, and distribution of the Python and R programming languages. This uses the Python 3.6 libraries. The environment was set up on the cloud using Intel Core(TM) i3 @3.4GHz 16GB of RAM and GeForce GTX 1060 GPU on CentOS 7. Keras and TensorFlow libraries were used as supporting libraries and code was written on Jupyter notebook which was set as part of anaconda. To set up the Python environment for running notebook is as h5py-2.9.0, Keras-2.2.4, NumPy-1.16.1, Tensorflow-1.12.0.

5-1- GRU Modelling for Text Classification

A GRU is an extension of a simple RNN which helps to counter the problem of the vanishing gradient by allowing the model to learn long-term dependencies in the structure of the text.

This paper focuses on classifying tweets using the Gated Recurrent Units method that makes use of the framework for embedding fixed size, matrix text. GRUs helps specifically inform the network of long-term dependencies. This is done by the addition of more variables into a basic RNN structure. The GRU layer uses the update gate to decide the amount of preceding information that should be passed on to the next activation while using the reset gate to determine the amount of preceding information that should be forgotten. The GRU update gate behaves similarly to an LSTM's forget gate and input gate. This decides what information you should throw away and what new information you should introduce. The GRU reset gate is another gate that is used to decide how much past information to forget.

GRU's have fewer tensor operations, so they're a little easier than LSTM's to practice. Which one is better, there is no definite winner.

This helps to ease the pressure of RNN in taking on long-term dependencies. For a long time, the GRU model integrates additional elements in the attestation process and becomes more complex while allowing further operations on secret GRU states. With no long-term dependencies, GRU will encode the variable-length series into a fixed-size representation, resulting in a lot of scalability for the model, and without any significant modification, it can be applied directly to longer contents, such as paragraphs, articles, etc.

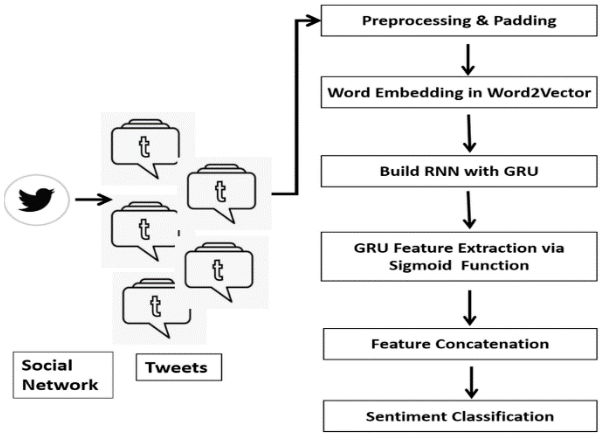


Figure 4: Architecture of the proposed text classifier based on RNN and GRU

To classify sentiments, we leveraged a set of 25000 reviews incorporating positive and negative labels. The encoding of pre-processed tweets is done through a sequence of integers which we sometimes also refer to as word indexes. The ordering of words is decided on their overall frequency in the data set. For example - The token or word is indexed based on frequency, for instance, 2. This indexing of words helps in shortlisting of words depending on their frequencies. Below is the sample code for the training dataset.

For several NLP activities, text representation plays a critical role. Successful word embedding will ease text encoding and boost the efficiency of classification. With that approach, different methods can represent the dataset.

The purpose of this study is to increase the efficiency in classification by combining the strength of different representations of words and different methods of deep learning. Furthermore, RNN and LSTM provide effective performance on various representations of the data. Although RNN is capable of capturing extraction of features in local regions as shown in Figure (4), GRU can extract good features from datasets with long-term dependencies such as natural languages and signals.

This allows the efficient use of RNN in datasets with near semantic relationships such as images. On the other hand, GRU delivers good performance on NLP issues and resolves semantic dependence among the words. Such approaches are useful because they can lead to the problems of classifying feelings. The results of the classification obtained confirmed the contribution. With word embedding methods like Word2Vec and Fast Text, some content disappears in the text [36].

Let's explain this through the example of preprocessing step text like URI data, emoticons, and meaningless grammatical words or stop words are discarded. The removed content still makes the part of the user through the process. To improve the dignity of the emotion one can mix multiple representations of text that helps to convey the honesty of the user's view. Such representations can contribute to various methods of deep learning and their extractions. Figure (5), reflects the deep learning model proposed in this architecture. The proposed model correlates with the embedding of CNN character-level while the second option works with active embedding LSTM Fast Text.

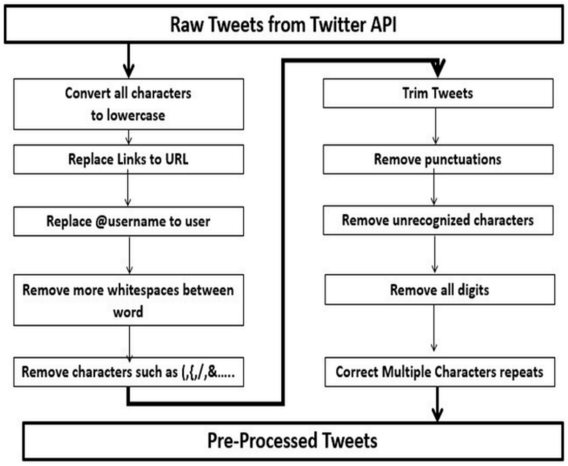

Raw Tweets from Twitter API
Convert all characters to lowercase
Replace Links to URL
Replace @username to user
Remove more whitespaces between word
Remove characters such as (,{,/,&.....
Trim Tweets
Remove punctuations
Remove unrecognized characters
Remove all digits
Correct Multiple Characters repeats
Pre-Processed Tweets

Figure 5: Pre-Processing Steps

The data generated on social media platforms by end-users contains a variety of contents such as slang, special characters, etc. along with standard alphabetic characters but many of these do not contribute to classification as shown in figure 5. For example, @usrname is a kind of neutral word not contributing to positive or negative text while performing sentiment analysis on social media platform posts such as Twitter. These texts contribute to noise in text processing problems [26]. Many available algorithms help increase classification performance by cleansing text data content. End users on social media platforms while posting their thoughts doesn't conform to any standard grammar rules of any spoken language and writes as per their thoughts leading to multilingual texts with many spell issues being posted. To overcome these difficulties in text processing we leveraged the Zemberek framework and performed multiple iterations to evaluate the impact of pre-processing our data set. These iterations helped increase classification performance.

6- Results & Discussion

One of the popular scenarios of Implementing natural language processing technique is Sentiment analysis whose aim is to identify whether a specific text conveys “positive” or “negative” sentiments

"The movie was so awesome”
“I slept peacefully for hours”

As a human reader, the comments in the above excerpt are understood to be of negative sentiment, but when it comes to machine analysis, one would need to create a machine learning model to classify sentiments. When leveraging the supervised learning approach, sampling needs to be done with multiple samples of text corpus with each corpus being labeled as positive or negative sentiment. Once this is done, the next step would be to create a machine learning model using this data.

Based on the example sentence above it is difficult for the machine learning model to classify it as positive or negative. If a tokenization or TFiDF approach is used, words such as 'glory' and 'caliber' could be misunderstood as conveying positive sentiments. Since there are no words interpreted as negative one would need to connect different parts of text structures to infer meaning out of the sentence. For Instance, the first sentence can be broken into two parts:

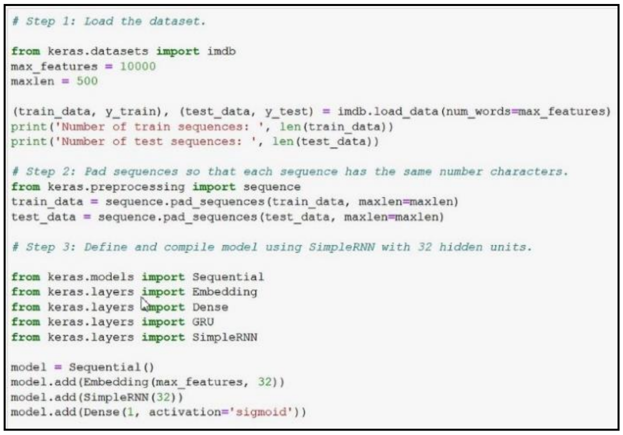
The first part of the sentence concludes the remark as positive. However, when the second sentence is considered its meaning somewhat infers negative sentiment. Hence need arises to retain long-term dependencies. . A simple RNN is, therefore, not good enough for the task. Let's try GRU for the sentiment classification task and see how it performs.

The Implementation is done on environment Implementing TensorFlow-2.3.1 and Keras-2.4.3. The execution is performed on a JupyterHub environment running python-3.8.3 on Conda with standard machine learning packages Installed. The specific

Keras libraries being used for text analysis are models and layers with Sequential, Embedding, Dense, GRU, and RNN.

6-1- Data Set Used

The dataset with 50K movie reviews is being utilized as an input for NLP and Text analytics and binary sentiment classification [41]. It comprises 25,000 highly polar movie reviews for training and testing purposes. For more information on the data set one can refer to this reference: http://ai.stanford.edu/~amaas/data/sentiment/



```
# Step 1: Load the dataset.

from keras.datasets import imdb
max_features = 10000
maxlen = 500

(train_data, y_train), (test_data, y_test) = imdb.load_data(num_words=max_features)
print('Number of train sequences: ', len(train_data))
print('Number of test sequences: ', len(test_data))

# Step 2: Pad sequences so that each sequence has the same number characters.
from keras.preprocessing import sequence
train_data = sequence.pad_sequences(train_data, maxlen=maxlen)
test_data = sequence.pad_sequences(test_data, maxlen=maxlen)

# Step 3: Define and compile model using SimpleRNN with 32 hidden units.

from keras.models import Sequential
from keras.layers import Embedding
from keras.layers import Dense
from keras.layers import GRU
from keras.layers import SimpleRNN

model = Sequential()
model.add(Embedding(max_features, 32))
model.add(SimpleRNN(32))
model.add(Dense(1, activation='sigmoid'))
```

Figure 6: Jupyterhub Notebook Snapshot of RNN Model

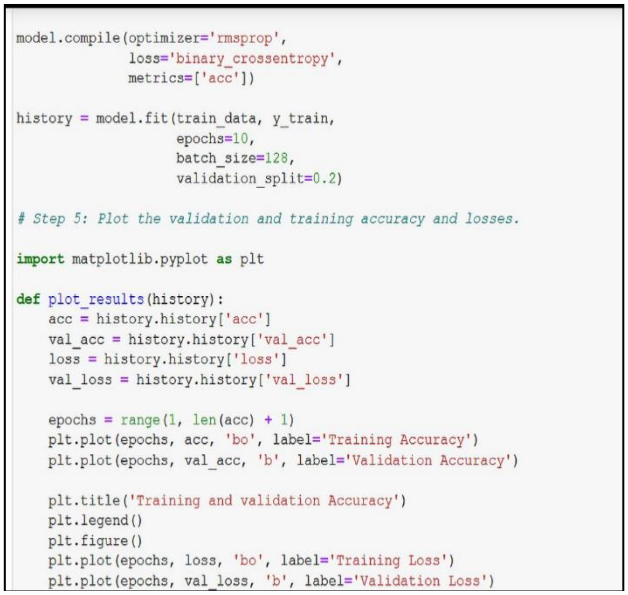
We can define maximum (max_) topmost occurring words while generating the sequence for training as 10,000. Sequence size can be restricted to 500. GRU unit can be used to build an RNN by importing necessary packages such as Sequential from Keras. layers import Embedding, from Keras. Sequential API of Keras is being used to build the model by importing the sequential model API from the Keras model as presented in figure(6). The embedding layer converts the input vector into a fixed-sized vector to be fed into the next layer of the network, if used, it must be added as the first layer to the network. The dense layer can be imported to give distribution over the target variable (0 or the embedding layer takes max_features as input, which is defined by us to be 10,000. The 32 value is set here as the next GRU layer expects 32 inputs from the embedding layer. . GRU unit can be imported to initialize the sequential model API and add the embedding layer, as follows:

```
model = Sequential()
model.add (Embedding (max_features, 32))
```

The embedding layer takes max_features as input, which is defined by us to be 10,000. The 32 value is set here as the next GRU layer expects 32 inputs from the embedding layer. Next, we'll add the GRU and the dense layer, as follows:

```
model.add (GRU (32))
model.add (Dense (1, activation='sigmoid'))
```

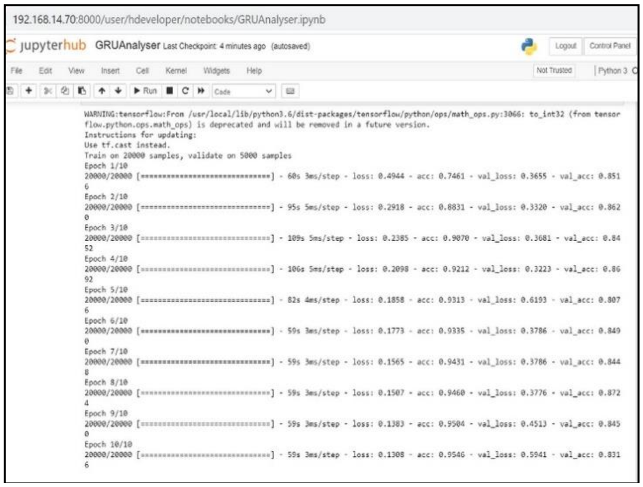
The fixed value integer 32 can be randomly chosen to function as one of the hyperparameters to tune when the network is designed, which also represents the dimensionality of the activation functions. The sigmoid function is used as an activation function as the dense layer only generates a single value which is considered as the probability of review and this is also our target variable. Note that we also assign 20% of the sample from the training data as the validation dataset. We also set the number of epochs to be 10 and the batch size to be 128 – that is, in a single forward-backward pass as shown in Figure (7).



```
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['acc'])

history = model.fit(train_data, y_train,
                    epochs=10,
                    batch_size=128,
                    validation_split=0.2)

# Step 5: Plot the validation and training accuracy and losses.

import matplotlib.pyplot as plt

def plot_results(history):
    acc = history.history['acc']
    val_acc = history.history['val_acc']
    loss = history.history['loss']
    val_loss = history.history['val_loss']

    epochs = range(1, len(acc) + 1)
    plt.plot(epochs, acc, 'bo', label='Training Accuracy')
    plt.plot(epochs, val_acc, 'b', label='Validation Accuracy')

    plt.title('Training and validation Accuracy')
    plt.legend()
    plt.figure()
    plt.plot(epochs, loss, 'bo', label='Training Loss')
    plt.plot(epochs, val_loss, 'b', label='Validation Loss')
```

Figure 7: Jupyterhub Snapshot of Model Validation and Plotting

The model is compiled with the binary cross-entropy loss and the rmsprop optimizer, to track the accuracy (train and validation) as the metric. Next, we fit the model on our sequence data. Note that we also assign 20% of the sample from the training data as the validation dataset.



192.168.14.70:8000/user/hdeveloper/notebooks/GRUAnalyser.ipynb
jupyterhub GRUAnalyser Last Checkpoint: 4 minutes ago (autosaved)
Logout
Control Panel
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Code

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensor
flow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 20000 samples, validate on 5000 samples
Epoch 1/10
20000/20000 [==============================] - 60s 3ms/step - loss: 0.4944 - acc: 0.7461 - val_loss: 0.3655 - val_acc: 0.851
6
Epoch 2/10
20000/20000 [==============================] - 95s 5ms/step - loss: 0.2918 - acc: 0.8831 - val_loss: 0.3320 - val_acc: 0.862
0
Epoch 3/10
20000/20000 [==============================] - 109s 5ms/step - loss: 0.2385 - acc: 0.9070 - val_loss: 0.3681 - val_acc: 0.84
52
Epoch 4/10
20000/20000 [==============================] - 106s 5ms/step - loss: 0.2098 - acc: 0.9212 - val_loss: 0.3223 - val_acc: 0.86
92
Epoch 5/10
20000/20000 [==============================] - 82s 4ms/step - loss: 0.1858 - acc: 0.9313 - val_loss: 0.6193 - val_acc: 0.807
6
Epoch 6/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1773 - acc: 0.9335 - val_loss: 0.3786 - val_acc: 0.849
0
Epoch 7/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1565 - acc: 0.9431 - val_loss: 0.3786 - val_acc: 0.844
8
Epoch 8/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1507 - acc: 0.9460 - val_loss: 0.3776 - val_acc: 0.872
4
Epoch 9/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1383 - acc: 0.9504 - val_loss: 0.4513 - val_acc: 0.845
0
Epoch 10/10
20000/20000 [==============================] - 59s 3ms/step - loss: 0.1308 - acc: 0.9546 - val_loss: 0.5941 - val_acc: 0.831
6
```

Figure 8: Jupyterhub Snapshot of epochs Classification on IMDB, [41] dataset.

When we mention validation split as a fit boundary while fitting deep learning model, its parts information into two sections for each epoch, for example, preparing information i.e., Training Data and validation data and since we are suing shuffle also it will rearrange dataset before spitting for that epoch. It prepares the model on training information and validates the model on validation data by checking its loss and precision

At the point when we are training the model in Keras, precision and loss in the Keras model for validating information could be variating with various cases. Ordinarily, with each epoch expanding, loss ought to be going lower and accuracy ought to be going higher as projected in Figure (8).

Epoch	Loss	Accuracy	Val Loss	Val Accuracy
1	0.4944	0.7461	0.3655	0.8510
2	0.2918	0.8831	0.3320	0.8620
3	0.2385	0.9070	0.3681	0.8400
4	0.2098	0.9212	0.3223	0.8600
5	0.1858	0.9313	0.6193	0.8070
6	0.1773	0.9335	0.3786	0.8490
7	0.1565	0.9431	0.3786	0.8440
8	0.1507	0.9460	0.3776	0.8720
9	0.1383	0.9504	0.4513	0.8450
10	0.1308	0.9546	0.5941	0.8310

Table 2: Evaluation metrics accuracy (higher is better) and Loss (lower is better).

GRU was used with word embedding's representation as Word2Vector in the experiment as a sequential input with RNNs models. We achieved the best classification accuracy of 87% in the 8th epoch as shown in Table (2).

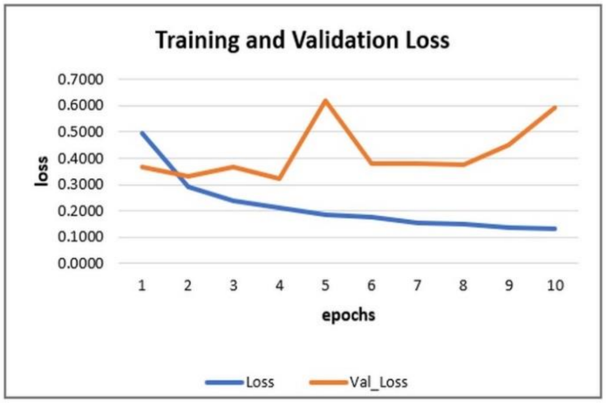


Figure 9: Training and Validation loss

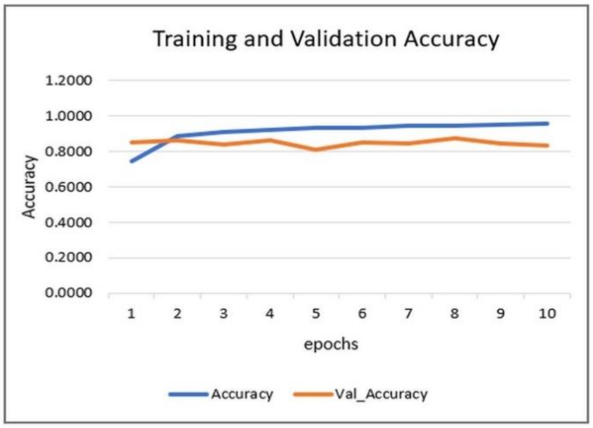


Figure 10: Training and Validation Accuracy

Figure (9) (10) defines the plot of accuracy and loss against epochs where accuracy achieved is directly proportional to values of epochs. Referring to Table (2) one cans infer easily that training set results on the model are comparable to epochs. The accuracy plot indicates that the model is underfitting before the point and overfit after the point. The loss diagram indicates underfitting before the Intersection point and overfitting after that.

7- Challenge

As per the copyright [21] survey by Zhang on sentiment analysis, sentiment analysis has the following granularity levels - document, sentence, and aspect wherein each level of classification add challenges based on granularity. A few of the key challenges involved in sentiment analysis are detection of sarcasm, negation, Word ambiguity, and Multipolarity. sarcasm is not only difficult to understand for a machine but also a human. The repetitive variation in words used in sarcastic sentences makes it hard to successfully train sentiment analysis models. Common topics, interests, and historical information must be shared between two people to make sarcasm available. Multiple approaches can be used for Sarcasm detection like Rule-based, Statistical, Machine Learning, and Deep Learning.
Negation detection can be done by marking as negated all the words from a negation cue to the next punctuation token. GRU and Long Short-Term Memory [9] can store information about a longer data sequence that has been processed.

One of the powerful techniques that have emerged in the recent past is Deep learning [34] which does its learning by using multiple layers of representations or data features and helps to produce state-of-the-art prediction results. There are manyfold challenges associated with traditional neural network techniques like gradient explosion and over-fitting [9], while deep GRU neural network model comes with low update efficiency and poor information processing capability among multiple hidden layers [21]

8- Conclusion and Future Scope

Sentiment analysis is one of the Important areas of natural language processing. Our paper presents the idea of sentiment classification of tweets and reviews using GRU. GRUs can use their update and reset gates to store and filter the information. This removes the issue of the vanishing gradient since the model does not wash out the new input every single time but preserves the relevant data and transfers it down to the network's next steps. Even in complex situations, if carefully trained, they may perform extremely well. We can conclude that gated recurrent units are a suitable model for sentiment analysis. Being a recurrent network, it can effectively capture long sequence data required for natural language understanding.

A GRU is an expansion of a basic RNN, which assists with combatting the vanishing gradient problem issue by permitting the model to learn long haul conditions in the content structure [2][8]. An assortment of utilization cases can profit from this compositional unit. a

State-of-the-art sentiment analysis and other NLP tasks could be created by leveraging GRU networks with 87% accuracy. In the future, we may have another progression over a basic RNN – Long Short-Term Memory (LSTM) network or Bi-GRU and Bi-LSTM Network as recommended in [18]. RNN with LSTM can analyze and predict based on the advantages of LSTM they carry with their new architecture.

References

[1] Andrew L , Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis." In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pp. 142-150. 2011.

[2] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5, no. 2, 1994,pp 157-166.

[3] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." IEEE transactions on pattern analysis and machine intelligence 35, no. 8, 2013, pp 1798-1828.

[4] B. Athiwaratkun, Ben, and Jack W. Stokes. "Malware classification with LSTM and GRU language models and a character-level CNN." In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2482-2486.

[5] Chen, Huimin, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. "Neural sentiment classification with user and product attention." In Proceedings of the 2016 conference on empirical methods in natural language processing, 2016, pp. 1650-1659.

[6] Chen, Peng, Zhongqian Sun, Lidong Bing, and Wei Yang. "Recurrent attention network on memory for aspect sentiment analysis." In Proceedings of the 2017 conference on empirical methods in natural language processing,2017 pp. 452-461.

[7] Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014 ." arXiv preprint arXiv:1406.1078.

[8] Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014." arXiv preprint arXiv:1412.3555.

[9] C.Zhou, Chunting, Chonglin Sun, Zhiyuan Liu, and Francis Lau. "A C-LSTM neural network for text classification."2015, arXiv preprint arXiv:1511.08630.

[10] Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch." Journal of machine learning research 12, no. ARTICLE 2011, pp 2493-2537.

[11] D. Tang, Duyu, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. "Learning sentiment-specific word embedding for twitter sentiment classification." In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics ,Volume 1, 2014 , pp. 1555-1565.

[12] Hemmatian, Fatemeh, and Mohammad Karim Sohrabi. "A survey on classification techniques for opinion mining and sentiment analysis." Artificial Intelligence Review 52, no. 3, 2019,pp 1495-1545.

[13] Guolong Liu, Xiaofei Xu, Bailong Deng, Siding Chen, and Li. "A hybrid method for bilingual text sentiment classification based on deep learning." In 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD),IEEE, 2016,pp. 93-98.

[14] Guan, Ziyu, Long Chen, Wei Zhao, Yi Zheng, Shulong Tan, and Deng Cai. "Weakly-Supervised Deep Learning for Customer Review Sentiment Classification." In IJCAI,2016, pp. 3719-3725.

[15] Goldberg, Yoav. "A primer on neural network models for natural language processing." Journal of Artificial Intelligence Research 57,2016, pp. 345-420.

[16] H.Wang, and Dequn Zhao. "Emotion analysis of microblog based on emotion dictionary and Bi-GRU." In 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), . IEEE, 2020, pp. 197-200.

[17]J. Pennington, R. Socher, and C. D. Manning. “Glove: Global vectors for word representation”. Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532-1543.

[18] Jabreel, Mohammed, Fadi Hassan, and Antonio Moreno. "Target-dependent sentiment analysis of tweets using bidirectional gated recurrent neural networks." In Advances in hybridization of intelligent methods, Springer, Cham, 2018, pp. 39-55.

[19] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, Yoshua Bengio. "Gated feedback recurrent neural networks." In International conference on machine learning, PMLR, 2015, pp. 2067-2075.

[20] Johnson, Rie, and Tong Zhang. "Effective use of word order for text categorization with convolutional neural networks," 2014, arXiv preprint arXiv:1412.1058.

[21] L Zhang, S Wang, B Liu. Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2018 Jul, 8(4):e1253.

[22[Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." In International conference on machine learning, PMLR, 2014, pp. 1188-1196.

[23]Li, Cheng, Xiaoxiao Guo, and Qiaozhu Mei. "Deep memory networks for attitude identification." In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining,2017, pp. 671-680.

[24] Luo LX. “Network text sentiment analysis method combining LDA text representation and GRU-CNN”, Personal and Ubiquitous Computing 23, no. 3 ,2019, pp 405-12.

[25]M Zulqarnain, R Ghazali, MG Ghouse.” Efficient processing of GRU based on word embedding for text classification”, International Journal of Informatics Visualisation, vol 3 No 4, 2019, pp.377-383.

[26] MU Salur, I Aydin. “A novel hybrid deep learning model for sentiment classification”. IEEE Access. 2020 Mar 23;8,pp 58080-93.

[27] Moraes, Rodrigo, João Francisco Valiati, and Wilson P. GaviãO Neto. "Document-level sentiment classification: An empirical comparison between SVM and ANN." Expert Systems with Applications 40, no. 2 , 2013, pp 621-633.

[28] Ö Yildirim. "A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification." Computers in biology and medicine 96 , 2018, pp189-202.

[29] R. Ni and H. Cao, "Sentiment Analysis based on GloVe and LSTM-GRU," In 2020 39th Chinese Control Conference (CCC), IEEE, 2020, pp. 7492-7497.

[30] S Yan, J Chai, L Wu. "Bidirectional GRU with Multi-Head Attention for Chinese NER." In 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), IEEE, 2020, pp. 1160-1164.

[31] S Yang, X Yu, Y Zhou. "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example." In 2020 International Workshop on Electronic Communication and Artificial Intelligence , IWECAI, IEEE, 2020, pp. 98-101.

[32] Tang, Duyu, Bing Qin, and Ting Liu. "Document modeling with gated recurrent neural network for sentiment classification." In Proceedings of the 2015 conference on empirical methods in natural language processing,2015, pp. 1422-1432.

[33]Tang, Duyu, Bing Qin, and Ting Liu. "Learning semantic representations of users and products for document level sentiment classification." In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing ,Volume 1, 2015, pp. 1014-1023.

[34]Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. "Improved semantic representations from tree-structured long short-term memory networks." 2015, arXiv preprint arXiv:1503.00075.

[35]Thet, Tun Thura, Jin-Cheon Na, and Christopher SG Khoo. "Aspect-based sentiment analysis of movie reviews on discussion boards." Journal of information science 36, no. 6 2010, pp 823-848.

[36]T. Mikolov, , Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. "Distributed representations of words and phrases and their compositionality."2013, arXiv preprint arXiv:1310.4546.

[37]Turian, Joseph, Lev Ratinov, and Yoshua Bengio. "Word representations: a simple and general method for semi-supervised learning." In Proceedings of the 48th annual meeting of the association for computational linguistics,2010, pp. 384-394.

[38]Y. Pan and M. Liang, "Chinese text sentiment analysis based on bi-gru and self-attention." In 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), . IEEE, 2020, vol. 1, pp. 1983-1988.

[39]Zhou, Zhiyuan, Yuan Qi, Zheng Liu, Chengchao Yu, and Zhihua Wei. "A C-GRU neural network for rumors detection." In 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), IEEE, 2018, pp. 704-708.

[40] Zhang, Wei, Quan Yuan, Jiawei Han, and Jianyong Wang. "Collaborative multi-Level embedding learning from reviews for rating prediction." In IJCAI, vol. 16, pp. 2986-2992. 2016.

[41] Data Set: Large Movie Review Dataset v1.0 by Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Available http://ai.stanford.edu/~amaas/data/sentiment/

Nidhi Chandra is a Research Scholar at Amity Institute of Information Technology and is a faculty at Department of Computer Science & Engineering, Amity University, Uttar Pradesh. She obtained her Master’s Degree in Computer Science from Guru Gobind Singh Indraprastha University, Delhi. With over 17 years of Academic experience and 2 years of Industry experience, her research interest lies in Natural Language Processing, Semantic Analysis and Data mining.

Prof. (Dr.)Laxmi Ahuja Ph.D. (CSE) working as Professor in Amity Institute of Information Technology with the role ranging from Lecturer to Professor to Dy. Director of the Department within a span of 21 years. in academics, her areas of interest include Search Engine, Data Mining and Soft Computing Approaches; published more than 100 research papers in International and National Conferences and Journals in SCOPUS Impact Factor Journals like Springer, Elsevier, Inderscience and several others. She has successfully filed number of patents under domain of Information Technology as an Inventor, which has been published in "International Journal of Patents" by Patent Department, Govt. of India. She has published many Lecture Notes for Springer Book chapter. Organized various sessions in International Conferences and Session chair in various Conferences. She has organized various Guest Lectures, Workshops and Faculty Development Programme. She is Associate Editor of Proceedings of the International Conference on Reliability, Infocom Technology and Optimization (Trends and Future Directions). She is actively involved in research activities. She has currently Eight research scholar under her guidance and supervised three students for Ph.D. She has also published number of Books related to operating system and Information Technology. She is Member of IEEE and Life Member of Computer Society of India, Past Vice chairman of IETE -The Institute of Electronics and Telecommunications Engineers, Senior member of IACSIT, Member of IEEE Society. She is actively involved in various activities of these societies. She is also Technical Chair member in many Conferences. She has Awarded as Guest of Honor in many Events.

Prof. (Dr.) Sunil Kumar Khatri is Director of Campus at Amity University in Tashkent, Uzbekistan. He is a Fellow of IETE, Sr. Life Member of IEEE, CSI and IASCSIT and Member of IAENG. He is Secretary in SREQOM, Convener, EAC, Past IEEE UP Section Executive Council and Past Vice-Chairman of CSI Noida Chapter. Dr. Sunil Kumar Khatri is Editor IJSAEM, Springer Verlag. He is in Editorial Board of several Journals from USA, Egypt, Hong Kong, Singapore, and India. He has twelve edited books, eleven guest edited special issues of international journals, eleven patents filed and more than 225 papers in international and national journals and proceedings. His areas of research are Artificial Intelligence, Software Reliability and Testing, and Data Analytics.

Prof. (Dr.) Himanshu Monga did Ph.D. from Thapar Institute of Engineering and Technology, (TU) Department Of ECE, Patiala, Punjab, INDIA, in the field of Optical networks / coding and Wireless networks. He did bachelor's degree in Electronics & Communication Engineering from Govt. Engineering College, Amravati University, Maharashtra, India and master's degree in Electronics and Telecommunication Engineering & Master's degree in management in Human resource management. Presently working as Professor & Head (ECE) at Jawaharlal Lal Nehru Government Engineering College, Sundernagar, Mandi (H.P). He is also having the charge of Dean Academics at JNGEC & OIC (Computer Centre). he has more than 21 years of work experience in academics and industry/research. He has published more than 140 Research papers in international/national conferences. Have more than 30 Thomson Reuter SCI publications to his credit. He is the life member of IEEE, ISTE and reviewer of many renowned journals. He has authored 08 Books/ chapters in renowned books. He has successfully completed 06 projects worth Rs 70 Lacks and successfully completed many consultancy projects. He has recently published 5 patents.

A New Game Theory-Based Algorithm for Target Coverage in Directional Sensor Networks

Elham Golrasan*
Faculty of Electrical and Computer Engineering, Malek Ashtar University of Technology, Iran
egolrasan@yahoo.com
Marzieh Varposhti
Department of Computer Engineering, Shahrekord University, Shahrekord, Iran
varposhti@aut.ac.ir

Received: 19/Oct/2020 Revised: 22/Mar/2021 Accepted: 15/Apr/2021

Abstract

One of the challenging problems in directional sensor networks is maximizing target coverage while minimizing the amount of energy consumption. Considering the high redundancy in dense directional sensor networks, it is possible to preserve energy and enhance coverage quality by turning off redundant sensors and adjusting the direction of the active sensor nodes. In this paper, we address the problem of maximizing network lifetime with adjustable ranges (MNLAR) and propose a new game theory-based algorithm in which sensor nodes try to adjust their working direction and sensing range in a distributed manner to achieve the desired coverage. For this purpose, we formulate this problem as a multiplayer repeated game in which each sensor as a player tries to maximize its utility function which is designed to capture the tradeoff between target coverage and energy consumption. To achieve an efficient action profile, we present a distributed payoff-based learning algorithm. The performance of the proposed algorithm is evaluated via simulations and compared to some existing methods. The simulation results demonstrate the performance of the proposed algorithm and its superiority over previous approaches in terms of network lifetime.

Keywords: Directional Sensor Networks; Target Coverage; Network Lifetime; Game Theory; Payoff-Based Learning Algorithm.

1- Introduction

Directional sensor networks (DSNs) contain several directional sensors deployed densely and randomly to cover a set of targets. Applications of such networks have been grown and widely applied in the field of industry and our daily life. In comparison with omni-directional sensor nodes, directional sensors have their unique characteristics, such as angle of view, working direction, and line of sight, therefore DSN applications require specific solutions for enhancing target coverage. Motility capability of a directional sensor node has a noticeable impact on the coverage enhancement in randomly deployed DSNs. Directional sensor nodes exploit motility to adjust their working direction. So motility can be used to minimize the overlapped regions. Because of limited energy resources in these networks, providing desired target coverage is a challenging problem [1-3].

In many applications, a large number of directional sensor nodes are randomly deployed in an area of interest. The availability of redundant sensors enhances the fault tolerance capability of the network. However, keeping all the sensor nodes active is not efficient because it leads to higher energy consumption. Therefore, one of the goal of this paper is to enhance the target coverage in a network of self-orienting sensor nodes. The other goal is decreasing the power consumption and increasing the network lifetime.

Since we are interested in the automated self-orienting of the nodes, we cast this problem as a non-cooperative game as it is a well-established tool for modeling coordination problems. There are many reasons to choose game theory as a method to solve the coverage problem in sensor networks. First, the principle of game theory allows sensor nodes to operate independently and calculate their proper orientation in a distributed manner. A well-designed gain function that takes into account all the limitations of the problem including the energy consumption, makes it possible to establish acceptable coverage in the sensor network. So, we can provide scalable network coverage using game theory. Finally, the game theory method is resistant to node failures and environmental disturbances [4].

In this paper, we model the coverage problem as a finite strategic game and propose a game theory-based algorithm (GT-based algorithm), in which the utility function is designed to capture the tradeoff between the worth of the

* Corresponding Author

covered area and the energy consumption due to sensing. An important issue is to devise distributed learning algorithms, using local information and processing abilities, to reach a Nash equilibrium (NE) of the game. To this end, we use a distributed payoff-based learning algorithm [5]. It has been proved that if all sensors (players) adhere to this algorithm, then each sensor selects the action profile that maximizes the total payoff of the sensors.

In most game theory based algorithm, a challenging problem is achieving the Nash equilibrium using a distributed learning algorithm [4, 6-10]. In [8, 10, 11], distributed learning algorithms are proposed to achieve NE in coverage problems. The authors in [12, 13] have studied distributed systems and propose distributed learning algorithms to achieve NE in potential games. However, there are two main drawbacks in this context: First of all, most of the results in this area have focused on converging to the NE, while in many cases it is not the optimum solution. Detecting this inefficiency is an extremely active research area in algorithmic game theory [14]. Secondly, it is often impossible to frame the interaction of a given system as a potential game [5]. We measure the performance of an action profile using the sum of the sensor's utility functions. Therefore, by designing the appropriate utility function for each player (sensor node) and applying a distributed payoff-based learning algorithm, coverage in the sensor network converges to a Pareto optimal action profile. The utility function is defined based on the tradeoff between coverage and energy consumption. Then each sensor learns how to adjust its working direction to maximize its utility function which corresponds to find its best orientation based on local information.

In the following, the main contributions of this study are presented:

- We formulate the maximum network lifetime with adjustable ranges (MNLAR) problem as a multiplayer repeated game in which each sensor as a player tries to maximize its utility function. The utility function is designed to capture the tradeoff between the worth of covered targets and the energy consumption due to sensing.
- We propose a distributed payoff-based learning algorithm that converges to an efficient action profile.
- The performance of the proposed algorithm is evaluated via simulations and compared to previous approaches. The simulation results show that the proposed algorithm results in activating the minimum number of directional sensors. In addition, active sensors learn how to adjust their sensing ranges to maximize the coverage. These bring about less energy consumption along with network lifetime extension.

The paper is organized as follows: In section 2, we briefly review recent studies in the context of sensor coverage problems. In section 3, we introduce the MNLAR problem in DSNs. Section 4 presents the proposed method and its formulation based on game theory. In section 5, simulation results are presented through several experiments. Finally, we conclude the paper in section 6.

2- Related Work

In this section, we briefly review the research work on coverage in wireless sensor networks. The coverage problem is usually divided into three categories: area coverage, point coverage, and barrier coverage [2]. The purpose of area coverage is to cover the whole area. Point coverage is the problem of covering Points of Interest (PoI) in the area. The barrier coverage guarantees that every movement that crosses a barrier of sensors will be detected.

Habibi et al. [15] have proposed a distributed Voronoi-based strategy to maximize the sensing coverage in a mobile sensor network. In this algorithm, each sensor moves through a gradient-based nonlinear optimization approach and is placed inside its Voronoi cell.

Ai et al. [16] have studied the problem of covering targets with directional sensors. They have formulated the problem as an optimization problem to maximize the coverage with a minimum number of sensors and proved that it is NP-complete. They have proposed several greedy heuristic methods to solve the problem.

Mohamadi et al [17] have proposed two Greedy-based algorithms for target coverage in directional sensor networks with adjustable sensing ranges. To maximize the sensor lifetime, they have used both scheduling and adjusting sensing range techniques to form cover sets to cover all targets in the network.

In [18], the authors have provided a GA-based algorithm to find cover sets of directional sensors with appropriate sensor ranges in order to solve the MNLAR (Maximum Network Lifetime with Adjustable Ranges) problem.

Yu et al. [19] have addressed the problem of providing K-coverage along with prolonging the network lifetime in wireless sensor networks with both centralized and distributed protocols. They have introduced a new concept of Coverage Contribution Area (CCA). Based on this concept, a lower sensor spatial density is provided.

The authors in [20] have designed a probabilistic coverage preservation protocol (CPP) to achieve energy efficiency and ensure a certain coverage rate. The purpose of the proposed protocol is to select the minimum number of probabilistic sensors to reduce energy consumption.

A graph model named Cover Adjacent Net (CA-Net) is proposed by Weng et al in [21] to simplify the problem of k-barrier coverage while reducing the complexity of computation. Based on the developed CA-Net, two distributed algorithms, called BCA and TOBA, are presented for energy balance and maximum network lifetime.

Mostafaei et al. [11] have proposed a distributed boundary surveillance (DBS) algorithm to cover the boundary and reduce the energy consumption of sensors. DBS selects the minimum number of sensors to increase the network lifetime using learning automata.

Li et al. [22] have proposed the Voronoi-based distribution approximation (VDA) algorithm. In the proposed algorithm, in order to maximize the coverage of the desired area, the most Voronoi edges are covered. In [23], the authors have proposed the distributed Voronoi-based self-redeployment algorithm (DVSA), aiming to improve the overall field coverage of mobile directional sensor networks. This paper has utilized the geometrical features of Voronoi diagram and the advantages of a distributed algorithm.

Recently, game-theoretic approaches have been taken into consideration to solve coverage problems in WSNs [4, 6, 24, 25]. In [26], the authors have proposed an algorithm based on game theory for the problem of maximizing coverage and reducing energy consumption. They have shown that the desired solution in this model is an NE strategy profile.

In [27], the authors have proposed a distributed learning method to maximize the area coverage in mobile directional sensor networks. Each sensor in collaboration with its neighbors tries to determine its best position and orientation.

The authors in [28] have considered the problem of area coverage without location information in mobile sensor networks. They have modeled this problem as a potential game and proposed a distributed learning algorithm to achieve an NE.

In [29] coverage of an unknown environment is investigated by robots. The state-based potential game was designed to control the robots' actions. The reward of sensing the areas and the penalty of energy consumption due to the sensors' movement are considered in the utility function. The sensors update their action profile using the Binary Log-Linear Learning (BLLL) in which the sensors must know an estimation of the outcome of their future actions. Hence, an estimation algorithm is proposed to assist the sensors in predicting the probability of targets in unknown areas. An improved EM algorithm is introduced to estimate the number of targets and other probability distribution parameters. In this study, we propose a game theory-based algorithm to optimally cover targets and reduce energy consumption.

3- Problem Definition

We consider a two-dimensional mission space, where n directional sensors with motility capability and a set of m target points, T, are initially located within a given area. We have defined several power levels, p, and a set of working directions, D, for sensor nodes. So, each sensor s_i has two parameters, working direction and power level. Sensor nodes can rotate around their axis and adjust their power level to cover a sector area. So, a directional sensor, s_i, monitors all targets within its sensing range and field of view. Each sensor has a limited energy resource. The amount of energy consumption is a function of sensing range; the greater sensing range, the more energy is consumed [30]. All the sensors in the network have the same characteristics in terms of initial battery power and energy consumption function.

Since increasing the power level is equivalent to increasing the sensing range which results in covering more targets, for each sensor direction $d_{i.j}$ and each power level $p > 1$, we have $T_{(d_{i.j}.\ b)} \subseteq T_{(d_{i.j}.\ p)}\ \forall b \in \{1.\ldots.p-1\}$, which $(d_{i,j}, p)$ is i^{th} sensor with j^{th} direction activated at power level of p. $T_{(d_{i.j}.\ b)}$ is the set of covered targets by sensor s_i. The power level p which is sufficient to cover target t_j by sensor $(d_{i.j}.\ p)$ is minimal if $t_j \in T_{(d_{i.j}.\ p)}$ and $t_j \notin T_{(d_{i.j}.\ b)}\ \forall b \in \{1.\ldots.p-1\}$. It means that target t_j cannot be covered by power level less than p. The notations used in this paper are listed in Table 1.

We assume the sensors are non-rechargeable. According to [17], the energy consumption due to displacement between the directions of a sensor is negligible, so it is ignored. Here, a positive parameter Δ^p is defined at each power level p [30]. The parameter Δ^p indicates the ratio between battery consumption at level p and level 1. Level 1 is the lowest and cheapest level. For example, if Δ^p=2, then the battery consumption at level p is twice that of level 1 (Δ^p=1).

Table 1: Notations

Notation	**Meaning**
n	Number of sensors
m	Number of targets
w	Number of working directions, $w \geq 1$
p	Number of different power levels, $p \geq 1$
s_i	i^{th} sensor, for all $i \in \{1.\ldots.n\}$
t_k	k^{th} target, for all $k \in \{1.\ldots.m\}$
l_i	Lifetime of sensor s_i
$d_{i,j}$	*J*th direction of *i*th sensor
D	Set of the working directions

S	Set of sensors, $\{s_1. \dots . s_n\}$
T	Set of targets, $\{t_1. \dots . t_m\}$
$(d_{i,j}, p)$	i^{th} sensor with j^{th} direction activated at power level of p
$T_{(di,j,\ p)}$	all the targets covered by i^{th} sensor with j^{th} direction at power level of p

Problem. How to divide the available sensors into cover sets so that each cover set covers all the targets in the area of interest with the goal of prolonging the network lifetime as much as possible. In other words, the main challenge of this process is assigning the appropriate working direction and sensing range to each sensor in a way that full target coverage and maximum network lifetime can be achieved.

For a better understanding, consider Fig. 1.A, which includes a scenario with three targets and four directional sensors to monitor the targets in the network. Fig. 1.B shows that each sensor has three directions $d_{i,j}$ ($1 \leq j \leq 3$) and two sensing ranges. Consider the set of sensor nodes with their best parameters that cover each target t_j. $C(t_1) = \{(d_{3.1}. 1). (d_{1.2}. 2)\}$, $C(t_2) = \{(d_{2.3}. 1). (d_{3.2}. 2)$, $C(t_3) = \{(d_{3.3}. 2). (d_{4.1}. 1)\}$. the purpose is to form the best cover set. The possible cover sets include:

$C_1 = \{(d_{1.2}. 2). (d_{2.3}. 1). (d_{3.3}. 2)\}$.

$C_2 = \{(d_{1.2}. 2). (d_{2.3}. 1). (d_{4.1}. 1)\}$.

$C_3 = \{(d_{1.2}. 2). (d_{3.2}. 2). (d_{4.1}. 1)\}$.

$C_4 = \{(d_{3.1}. 1). (d_{2.3}. 1). (d_{4.1}. 1)\}$.

Therefore, in this example, the cover set C_4 is more desirable because of less energy consumption.

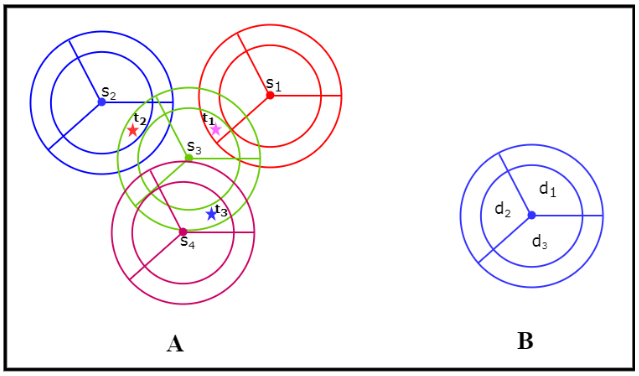


Fig. 1. (A) Example network with four directional sensors and three targets. (B) A directional sensor with three directions and two sensing ranges.

4- Proposed GT-Based Algorithm

In this section, we propose a game theory-based algorithm (GT-based algorithm) to target coverage. The new method is a solution to the MNLAR problem in DSNs. It converges to an efficient action profile using a distributed payoff-based learning algorithm. The output of the proposed algorithm is a cover set containing sensors with appropriate sensing ranges and working directions that can monitor all targets within the network. To calculate the activation time of the constructed cover set, we consider the sensor that minimizes $\frac{L_i}{\Delta p}$. Then the residual energy of the sensors in the cover set is calculated. The sensors that have no residual energy are removed from the list of available sensors. The GT-based algorithm continues its operation until all targets are fully covered. Finally, the sum of the activation times is returned as the network lifetime.

4-1- Background in Game Theory

In this section, we consider a brief review of the concepts in game theory. More information about game theory and learning in game theory is mentioned in [31, 32].

The strategic game $G \coloneqq \langle N. A. U \rangle$ has three components:

The finite set of N players (sensors) where $N \coloneqq \{1. \dots . n\}$.

An action set $A = A_1 \times \dots \times A_n$ where A_i is the finite action set of player i.

The set of utility (payoff) functions U. where the utility function $U_i\colon A \rightarrow R$ models the benefit of player i over action profiles.

For an action profile $a = (a_1. a_2. \dots . a_n) \in A$. $a_{-i} = \{a_1. \dots . a_{i-1}. a_{i+1}. \dots . a_n\}$ denotes the action profile of all players other than player i. Therefore, the action profile a can be represented as $(a_i. a_{-i})$.

The welfare of an action profile $a \in A$ is defined as follows:

$$W(a) = \sum_{i \in N} u_i(a) \tag{1}$$

if an action profile $a \in A$ maximizes the welfare, then the action profile a is efficient. In other words:

$$a \in \arg max_{a' \in A} W(a') \tag{2}$$

4-2- Game Formulation

Suppose *m* targets $T = \{t_1. t_2. \dots . t_m\}$ located in known locations in the area. A set of directional sensors $S = \{s_1. s_2. \dots . s_n\}$ with adjustable sensing ranges are deployed adjacent to the targets to completely monitor them. Sensors are static with variable sensing ranges between r_{min} and r_{max}. We assume that the communication range of each sensor *i* (R_i) is at least twice the r_{max} ($R_i \geq 2r_{max}$). Thus, each sensor can transmit state information to its neighbors and interact with its neighbors.

The worth of each target $t_k (1 \leq k \leq m)$ is denoted with $f_k \geq 0$. Each sensor s_i selects its mode from the set $Q_i = \begin{cases} 1 & if\ sensor\ i\ is\ on \\ 0 & if\ sensor\ i\ is\ off \end{cases}$.

The sensor's direction is determined by d_i and the set $D_i = \{d_{i.1}. d_{i.2}. \dots . d_{i.w}\}$ includes the defined working directions of sensor nodes. Each sensor s_i chooses its sensing range r_i from the discrete set $R_i = \{r_{i.1}. \dots . r_{i.p}\}$. The action of each sensor (player) s_i is shown by a vector a_i and defined as follows: $a_i = (c_i. d_i. r_i) \in A_i = Q_i \times D_i \times R_i$.

As mentioned before, $T(d_{i.j}. a)$ is a set of targets covered by sensor direction $d_{i.j}$ while its power level is a. For each target $t_k \in T(d_{i.j}. a)$, $n_k(s)$ represents the number of sensors that can observe the target point *k* and is defined as follows:

$$n_k(s) = \sum\nolimits_{i \in n} c_i \,;\ t_k \in T(d_{i.j}. a) \tag{3}$$

The profit of observing the target point t_k, which is shown by f_k , is evenly divided by the sensors that observe t_k. So, the utility that sensor s_i obtains due to sensing is equal to

$$c_i(\sum_{t_k \in T(d_{i.j}.a)} \frac{f_k}{n_k(s)}) \tag{4}$$

Due to energy constraints, we consider the energy consumption parameter in the utility function. We assume that the energy consumption of sensor nodes is because of their sensing activity. So, the energy consumption of each sensor node depends on its sensing range and is defined as (5).

$$E_i^{sense}(a_i) = k_i c_i (r_i)^2 \tag{5}$$

in which $k_i > 0$is a weighting factor related to energy consumption. Therefore, the utility function of the sensor s_i represents its contribution to the coverage task and energy consumption due to sensing. We consider the utility function for sensor s_i as follows:

$$u_i(a) = c_i \left(\sum_{t_k \in T(d_{i.j}.a)} \frac{f_k}{n_k(s)} \right) - E_i^{sense}(a_i) \tag{6}$$

In the following, we present a new distributed learning algorithm that leads to Pareto optimal outcomes.

4-3- Payoff-Based Learning Algorithm

The game G is repeated each time $t \in \{0.1.2. \dots\}$. In time stamp *t*, the sensors simultaneously select their actions, so the action profile is $a(t) = (a_1(t). \dots . a_n(t))$ and each sensor s_i receives utility $u_i(a(t))$. The sensor s_i will select the action $a_i(t)$ according to the probabilistic distribution $p_i(t) \in \Delta(A_i)$. $p_i(t)$ represents the strategy of sensor s_i at time *t*. $p_i^{a_i}(t)$ indicates the probability that the sensor s_i selects action $a_i(t)$ at time *t* according to the strategy $p_i(t)$. The sensor's strategy at time *t* depends on observations in previous times $\{0.1.2. \dots . t-1\}$.

The strategies of the sensors are updated by the information they have gathered. We know that the sensors here have limited observations. In this situation, sensors must learn to play an action profile that maximizes welfare. In this case, the sensors only have access to the actions they played and the utilities they received. Therefore, the strategy adjustment mechanism of sensor s_i is as follows:

$$p_i(t) = F_i(\{a_i(\tau). u_i(a(\tau))\}_{\tau = 0.1. \dots . t-1}) \tag{7}$$

Such an algorithm is called payoff-based or completely uncoupled [33]. It is proved in [5, 34-37] that for finite strategic games, there are completely uncoupled learning rules that lead to Pareto optimal Nash equilibria. We use the learning rule presented in [5]. This distributed learning rule leads to the convergence of the game into a Pareto optimal action profile, which maximizes the welfare.

Each sensor has a baseline action and a baseline utility, that is expected to play and receive, respectively. Each sensor has an internal state variable called mood. Mood defines the sensor behavior as follows. There are two distinct types of moods: content and discontent. When a sensor is in the content mood, baseline action is selected with high probability. When a sensor is in a discontent mood, an action differs from the baseline action is selected with a high probability. Each sensor updates its mood after choosing an action and receiving a payoff by comparing the action played and the payoff received with its baseline action and baseline payoff. At each time step, the sensor's state is represented by a triple $[\bar{a}_i. \bar{u}_i. m_i]$, where

- $\bar{a}_i \in A_i$ is the baseline action.
- $\bar{u}_i$ is the baseline utility.
- m_i is the mood of sensor *i*, which can be content (C) or discontent (D).

The main steps of the distributed learning algorithm for Pareto optimality are described as follows:

Step 1- Initialization:

- At stage $t = 0$, each player randomly selects and plays any action, $a_i(0)$.
- This action will be initially set as the player's baseline action at stage 1, $\bar{a}_i(1) = a_i(0)$.
- Likewise, the player's baseline utility at stage 1 is initialized as $\bar{u}_i(1) = u_i(a(0))$.

- the player's mood at stage 1 is set as $m_i(1) = D$.

Step 2- Action Selection:
At each subsequent stage $t > 0$, each player selects his action according to the following rules.

- If the mood of sensor *i* is content, i.e., $m_i(t) = C$, the sensor chooses an action $a_i(t)$ according to the following probability distribution
$$p_i^{a_i}(t) = \begin{cases} \dfrac{\varepsilon^c}{|A_i| - 1} & for\ a_i \neq \bar{a}_i \\ 1 - \varepsilon^c & for\ a_i = \bar{a}_i \end{cases}$$
where $|A_i|$ represents the cardinality of the set A_i and *c* is a constant that satisfies $c > n$.
- If the mood of sensor *i* is discontent, i.e., $m_i(1) = D$, the sensor chooses an action a_i according to the following probability distribution
$$p_i^{a_i}(t) = \frac{1}{|A_i|} \quad for\ every\ a_i \in A_i$$
Note that the benchmark action and utility play no role in the sensor dynamics when the sensor is discontent.

Step 3- Baseline Action, Baseline Utility, and Mood Update:

Each sensor *i* sends its status information including $a_i(0).F(s_i)$ to its neighboring nodes or nodes that are less than or equal to twice the sensing range of the node *i*. Then each sensor *i* computes the payoff $u_i\big(a_i(t).a_{-i}(t)\big)$ based on the data collected from neighbors. the state is updated according to the following rules.

- First, the baseline action and baseline utility at stage *t+1* are set as
$\bar{a}_i(t+1) = a_i(t)$
$\bar{u}_i(t+1) = u_i\big(a_i(t).a_{-i}(t)\big)$
- The mood of sensor *i* is updated as follows.
$$if \begin{bmatrix} \bar{a}_i(t) \\ \bar{u}_i(a(t)) \\ m_i(t) \end{bmatrix} = \begin{bmatrix} a_i(t) \\ u_i(a(t)) \\ C \end{bmatrix} then\ m_i(t) = C$$
Else
$$m_i(t+1) = \begin{cases} C & with\ probability\ \varepsilon^{1-u_i(a(t))} \\ D & with\ probability\ 1 - \varepsilon^{1-u_i(a(t))} \end{cases}$$

Step 4- Return to Step 2 and repeat.

The learning algorithm produces a sequence of action profiles $a(1).\dots.a(t)$, in which the behavior of a sensor *i* in each time $t = 1.2.\dots$ depends on the baseline action $\bar{a}_i$, the baseline utility $\bar{u}_i$, and the mood $m_i(t) \in \{C.D\}$. To converge the game into an efficient action profile, the game's structure must be interdependent [35]. In the following, the definition of interdependence is fully described.

Definition1 (Interdependence): A finite game G is interdependent If for any action profile $a \in A$ and any appropriate subset of the sensors $J \subset N$, There is a sensor $i \notin J$ and a selection of actions $a'_J \in \prod_{j \in J} A_j$ so that $u_i(a'_J.a_{-J}) \neq u_i(a_J.a_{-J})$.
In general, the interdependence condition states that the sensors cannot be divided into two distinct subsets that do not interact with one another. For this reason, we assume that the sensors are deployed in the area so that they cannot be divided into two distinct subsets and the condition of interdependence is established in the game.

Theorem1: Consider a finite interdependent game with *n* players. Under the distributed learning algorithm for Pareto optimization defined above, a state $z = [\bar{a}.\bar{u}.m] \in Z$ is stable if and only if the following conditions are met.

- The action profile $\bar{a}$ optimizes the welfare $W(\bar{a}) = \sum_{i \in N} u_i(\bar{a})$.
- The baseline actions and payoffs are aligned, i.e. for each sensor *i*, $\bar{u}_i = u_i(\bar{a})$.
- All sensors are in the content mood i.e. for each sensor *i*, $m_i = C$.

Proof of the theorem depends on the resistance trees for Markov's decision process, and it has been proven in [38].

5- Simulation Results

In this section, we evaluate the performance of the proposed algorithm through several experiments. The algorithms are simulated on MATLABR2017b and implemented on a system with an Intel Core i7 processor, 3.4 GHz CPU, and 4 GB RAM. The most important criterion for evaluating the performance of the algorithm is the network lifetime. The network lifetime is defined as the time that the sensor nodes can cover all of the targets. Each experiment examines the impact of different parameters on the network lifetime. Here, to model a DNS, *m* targets and *n* directional sensors are deployed randomly and uniformly in a $500(m) \times 500(m)$ area. Each sensor has several working directions and sensing ranges. For each sensor node, only one working direction and one sensing range can be activated at each unit of time. By default, the number of sensors and targets are 100 and 10, respectively. Also, we have considered 3 working directions for each sensor. The sensing range of each sensor can be adjusted from 80-120(m) with incremental step 10(m). We assume that each sensor initially has one unit of energy.
To establish the interdependence condition, those targets that are not monitored by any sensor direction are ignored and all sensors that cannot cover any targets are removed from the list of sensors. The simulation parameter is chosen as $k_i = 3 \times 10^{-5}. f_q = 0 \cdot 1. c = 100$. According to [38], employing an annealing schedule $\varepsilon_t = \frac{1}{\sqrt{t}}$ in the learning algorithm guarantees the convergence to an efficient action profile.

Experiment 1. This experiment is performed to provide an intuitive example for the implementation of the proposed algorithm. Five targets (m=5) and 20 directional sensors (n=20) are randomly deployed in the area. Fig. 2 shows the initial configuration of the network, and Fig. 3 shows the final configuration of the network after 10^5 iterations. According to Fig. 3, in addition to full coverage of the targets, energy consumption in the area has decreased due to the adjustment of working direction and sensing range of all sensor nodes. In this figure, all targets are covered with the minimum number of sensors with the least overlapping area (each target is exactly in one sector). Consequently, it is clear that the result is a Pareto optimal action profile. Fig. 4 presents the evolution of the welfare for $\varepsilon_t = \frac{1}{\sqrt{t}}$. The result shows the convergence of the welfare function to its maximum value. The reason is that energy consumption due to sensing is considered in the utility function.

Experiment 2. This experiment is designed to evaluate the relationship between the number of directional sensors and the lifetime of the network. To this end, we have considered sensor networks with 60-100 sensor nodes. Fig. 5 shows that increasing the number of sensor nodes results in enhancing the network lifetime. The reason is that each target is covered with more sensors, so more cover sets are constructed and the network lifetime is increased. Simulation results demonstrate that as the number of sensors increases, the proposed GT-based algorithm performs better than the Genetic-based algorithm [18] and Greedy-based algorithm [17]. This is due to the iterative property of learning algorithms and the more efficient management of energy consumption in the proposed game.

Experiment 3. This experiment is performed to determine how the number of targets affects the network lifetime. For this purpose, we have considered an area of interest with 6-14 target points. According to Fig. 6, network lifetime decreases when the number of targets increases. The reason is that as the number of targets increases, more sensors are needed to monitor them. This will cause the sensors to run out of energy earlier, so results in reducing the network lifetime.

Fig. 7 shows the energy consumption of the cover sets in Greedy-based, Genetic-based, and GT-based algorithms based on the number of targets. As expected, an increase in the number of targets increases energy consumption. The reason is that more sensors are needed to cover more targets. The results confirm that the GT-based algorithm consumes less energy compared to the other two algorithms since the proposed algorithm activates fewer sensor nodes, so the energy consumption due to sensing in GT-based algorithm is less than Genetic-based algorithm and Greedy-based algorithm.

Experiment 4. This experiment is performed to investigate the effect of the sensing range on the network lifetime. The sensing range is fixed between 80 and 120 with incremental step 10. According to the results presented in Fig. 8, an increase in the sensing range leads to an improvement in the network lifetime. This is because of this fact that increasing the sensing range results in covering more targets, so fewer sensors are needed to cover all targets. In comparison with the other two algorithms, experiment results confirm that the proposed algorithm is more successful in terms of maximizing network.

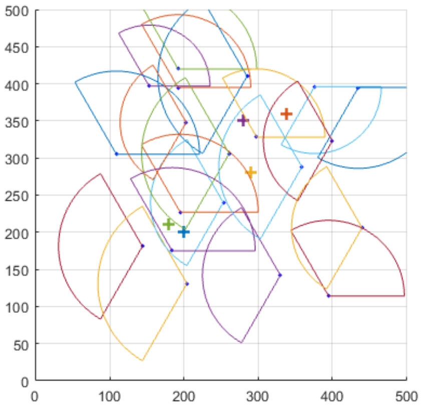


Fig. 2. The initial configuration of the network where "•" and "+" are sensor nodes and targets, respectively.

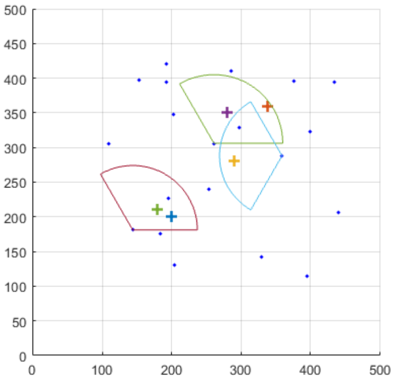


Fig. 3. The final configuration of the network after 10×10^4 iterations.

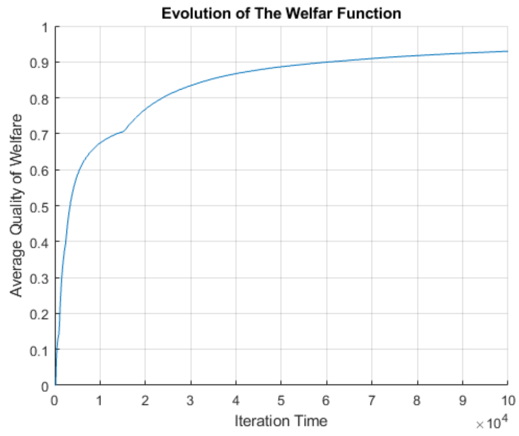


Fig. 4. The evaluation of the welfare function in $\varepsilon_t = \frac{1}{\sqrt{t}}$.

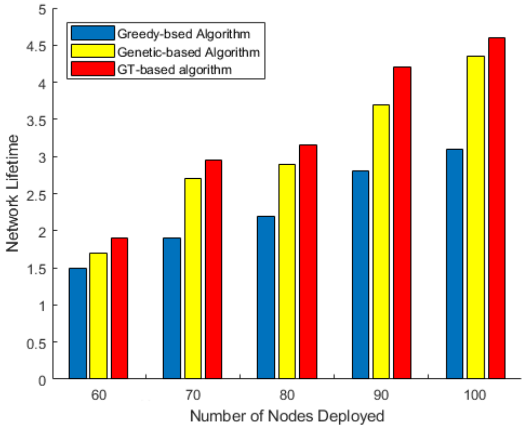


Fig. 5. Effect of the number of sensors on the network lifetime.

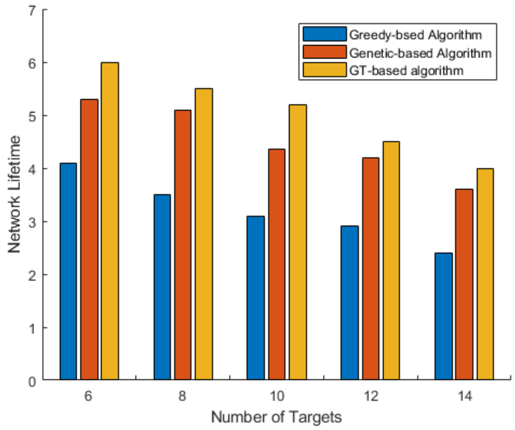


Fig. 6. Effect of the number of targets on the network lifetime.

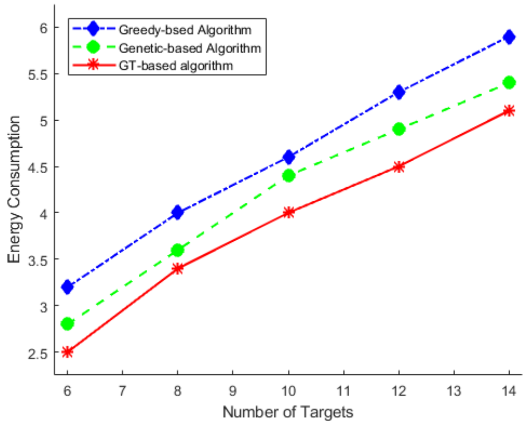


Fig. 7. Effect of the number of targets on energy consumption.

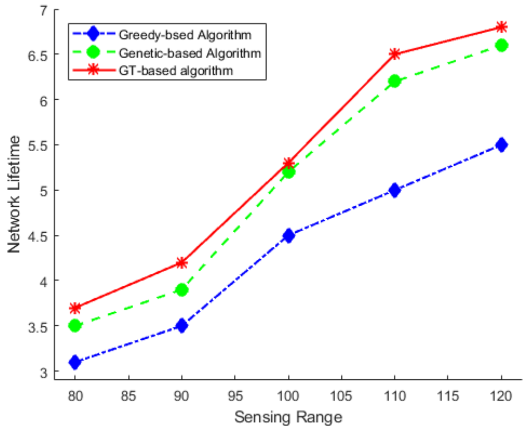


Fig. 8. Effect of sensing range on the network lifetime.

6- Conclusion

In this paper, we presented a new game theory-based algorithm for target coverage in networks containing sensors with multiple directions and sensing ranges to extend network lifetime. Due to the energy limitations in sensor networks, we formulate the target coverage problem as a finite strategic game in which a utility function is formulated to consider the tradeoff between energy consumption and coverage quality. To achieve an efficient action profile, we present a distributed payoff-based learning algorithm. The performance of our proposed algorithm was evaluated via simulations and compared to the greedy-based and genetic-based algorithms. The simulation results demonstrated the performance of our proposed algorithm and its superiority over previous approaches in terms of increasing the network lifetime.

References

[1] B. D. Deebak and F. Al-Turjman, "A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks," *Ad Hoc Networks,* vol. 97, p. 102022, 2020.

[2] K. Evbarunegbe, "Coverage in Wireless Sensor Networks: A Survey," Texas A&M University-Kingsville, 2020.

[3] S. Harizan and P. Kuila, "Evolutionary algorithms for coverage and connectivity problems in wireless sensor networks: a study," in *Design Frameworks for Wireless Networks*: Springer, 2020, pp. 257-280.

[4] S. Rahili, J. Lu, W. Ren, and U. M. Al-Saggaf, "Distributed Coverage Control of Mobile Sensor Networks in Unknown Environment Using Game Theory: Algorithms and Experiments," *IEEE Transactions on Mobile Computing,* vol. 17, no. 6, pp. 1303-1313, 2018.

[5] J. R. Marden, H. P. Young, and L. Y. Pao, "Achieving pareto optimality through distributed learning," *SIAM Journal on Control and Optimization,* vol. 52, no. 5, pp. 2753-2770, 2014.

[6] T. AlSkaif, M. G. Zapata, and B. Bellalta, "Game theory for energy efficiency in wireless sensor networks: Latest trends," *Journal of Network and Computer Applications,* vol. 54, pp. 33-61, 2015.

[7] E. Golrasan, H. SHIRAZI, and K. Dadashtabar Ahmadi, "Sensing Radius Adjustment in Wireless Sensor Networks: A Game Theoretical Approach," *INTERNATIONAL JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY RESEARCH,* vol. 11, no. 4, pp. -, 2019.

[8] F. Hajjej, M. Hamdi, R. Ejbali, and M. Zaied, "A distributed coverage hole recovery approach based on reinforcement learning for Wireless Sensor Networks," *Ad Hoc Networks,* vol. 101, p. 102082, 2020.

[9] J. R. Marden and J. S. Shamma, "Game theory and control," *Annual Review of Control, Robotics, and Autonomous Systems,* vol. 1, pp. 105-134, 2018.

[10] N. Mehendale, "Game Theory-Based Planning of Nodes in Wireless Sensor Networks for Optimum Coverage With Maximum Battery Life," *Available at SSRN 3639170,* 2020.

[11] H. Mostafaei, M. U. Chowdhury, and M. S. Obaidat, "Border surveillance with WSN systems in a distributed manner," *IEEE Systems Journal,* vol. 12, no. 4, pp. 3703-3712, 2018.

[12] N. Li and J. R. Marden, "Designing games for distributed optimization," *IEEE Journal of Selected Topics in Signal Processing,* vol. 7, no. 2, pp. 230-242, 2013.

[13] B. S. Pradelski and H. P. Young, "Learning efficient Nash equilibria in distributed systems," *Games and Economic behavior,* vol. 75, no. 2, pp. 882-897, 2012.

[14] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge University Press, 2007.

[15] J. Habibi, H. Mahboubi, and A. G. Aghdam, "A gradient-based coverage optimization strategy for mobile sensor networks," *IEEE Transactions on Control of Network Systems,* vol. 4, no. 3, pp. 477-488, 2016.

[16] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization,* vol. 11, no. 1, pp. 21-41, 2006.

[17] H. Mohamadi, S. Salleh, and M. N. Razali, "Heuristic methods to maximize network lifetime in directional sensor networks with adjustable sensing ranges," *Journal of Network and Computer Applications,* vol. 46, pp. 26-35, 2014.

[18] A. Alibeiki, H. Motameni, and H. Mohamadi, "A new genetic-based approach for maximizing network lifetime in directional sensor networks with adjustable sensing ranges," *Pervasive and Mobile Computing,* vol. 52, pp. 1-12, 2019.

[19] J. Yu, S. Wan, X. Cheng, and D. Yu, "Coverage contribution area based $ k $-coverage for wireless sensor networks," *IEEE Transactions on Vehicular Technology,* vol. 66, no. 9, pp. 8510-8523, 2017.

[20] J.-P. Sheu and H.-F. Lin, "Probabilistic coverage preserving protocol with energy efficiency in wireless sensor networks," in *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, 2007, pp. 2631-2636: IEEE.

[21] C.-I. Weng, C.-Y. Chang, C.-Y. Hsiao, C.-T. Chang, and H. Chen, "On-supporting energy balanced $ k $-barrier coverage in wireless sensor networks," *IEEE Access,* vol. 6, pp. 13261-13274, 2018.

[22] L. Jing, W. Ruchuan, H. Huang, and S. Lijuan, "Voronoi-based coverage optimization for directional sensor networks," *Wireless Sensor Network,* vol. 1, no. 05, p. 417, 2009.

[23] T.-W. Sung and C.-S. Yang, "Distributed Voronoi-based self-redeployment for coverage enhancement in a mobile directional sensor network," *International Journal of Distributed Sensor Networks,* vol. 9, no. 11, p. 165498, 2013.

[24] J. R. Marden and J. S. Shamma, "Game Theory and Control," *Annual Review of Control, Robotics, and Autonomous Systems,* no. 0, 2017.

[25] S. Rahili, "Distributed Optimization in Multi-Agent Systems: Game Theory Based Sensor Coverage and Continuous-Time Convex Optimization," UC Riverside, 2016.

[26] M. Movassagh and H. S. Aghdasi, "Game theory based node scheduling as a distributed solution for coverage control in wireless sensor networks," *Engineering Applications of Artificial Intelligence,* vol. 65, pp. 137-146, 2017.

[27] M. Varposhti, P. Saleh, S. Afzal, and M. Dehghan, "Distributed area coverage in mobile directional sensor networks," in *2016 8th International Symposium on Telecommunications (IST)*, 2016, pp. 18-23: IEEE.

[28] M. Varposhti, V. Hakami, and M. Dehghan, "Distributed coverage in mobile sensor networks without location information," *Autonomous Robots,* vol. 44, no. 3, pp. 627-645, 2020.

[29] M. Hasanbeig and L. Pavel, "Distributed Coverage Control by Robot Networks in Unknown Environments Using a Modified EM Algorithm," *World Academy of*

Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 11, no. 7, pp. 721-729, 2017.

[30] R. Cerulli, R. De Donato, and A. Raiconi, "Exact and heuristic methods to maximize network lifetime in wireless sensor networks with adjustable sensing ranges," *European Journal of Operational Research,* vol. 220, no. 1, pp. 58-66, 2012.

[31] J. R. Marden, "Learning in large-scale games and cooperative control," CALIFORNIA UNIV BERKELEY DEPT OF MECHANICAL ENGINEERING2007.

[32] H. P. Young, *Individual strategy and social structure: An evolutionary theory of institutions*. Princeton University Press, 2001.

[33] D. P. Foster and H. P. Young, "Regret testing: learning to play Nash equilibrium without knowing you have an opponent," *Theoretical Economics,* vol. 1, no. 3, pp. 341-367, 2006.

[34] I. Ariel and Y. Babichenko, *Average testing and the efficient boundary*. Center for the study of Rationality, 2011.

[35] D. Fudenberg and E. Maskin, "The folk theorem in repeated games with discounting or with incomplete information," in *A Long-Run Collaboration On Long-Run Games*: World Scientific, 2009, pp. 209-230.

[36] H. P. Young, "Learning by trial and error," *Games and economic behavior,* vol. 65, no. 2, pp. 626-643, 2009.

[37] H. P. Young and B. S. Pradelski, "Learning Efficient Nash Equilibria in Distributed Systems," 2010.

[38] A. Menon and J. S. Baras, "Convergence guarantees for a decentralized algorithm achieving Pareto optimality," in *2013 American Control Conference*, 2013, pp. 1932-1937: IEEE.

Elham Golrasan received her M.Sc. degree in Computer Engineering from Sharif University of Technology, Tehran, Iran, in 2013 and her PhD degree from Malek-Ashtar University of Technology, Tehran, Iran, in 2021. Her research interests include Wireless Sensor Networks, Game Theory and Learning Algorithms.

Marzieh Varposhti received her B.Sc. and M.Sc. from Isfahan University, Isfahan, Iran in 2005 and 2008 respectively, and Ph.D. from Amirkabir University of Technology (AUT), Tehran, Iran in 2015. She is currently an assistant professor in the department of computer engineering at Shahrekord University. Her research interests are in wireless networks, sensor networks, and IOT.

Optimal Clustering-based Routing Protocol Using Self-Adaptive Multi-Objective TLBO For Wireless Sensor Network

Ali Sedighimanesh
Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran
ali.sedighimanesh@gmail.com
Hessam Zandhessami*
Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran.
Zandhessami@srbiau.ac.ir
Mahmood Alborzi
Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran.
Mahmood_alborzi@yahoo.com
Mohammadsadegh Khayyatian
Institute for science and technology studies, shahid Beheshti university, Tehran, Iran.
M_khayatian@sbu.ac.ir

Received: 28/Nov/2020 Revised: 27/Mar/2021 Accepted: 22/Apr/2021

Abstract

Wireless sensor networks consist of many fixed or mobile, non-rechargeable, low-cost, and low-consumption nodes. Energy consumption is one of the most important challenges due to the non-rechargeability or high cost of sensor nodes. Hence, it is of great importance to apply some methods to reduce the energy consumption of sensors. The use of clustering-based routing is a method that reduces the energy consumption of sensors. In the present article, the Self-Adaptive Multi-objective TLBO (SAMTLBO) algorithm is applied to select the optimal cluster headers. After this process, the sensors become the closest components to cluster headers and send the data to their cluster headers. Cluster headers receive, aggregate, and send data to the sink in multiple steps using the TLBO-TS hybrid algorithm that reduces the energy consumption of the cluster heads when sending data to the sink and, ultimately, an increase in the wireless sensor network's lifetime. The simulation results indicate that our proposed protocol (OCRP) show better performance by 35%, 17%, and 12% compared to ALSPR, CRPD, and COARP algorithms, respectively. Conclusion: Due to the limited energy of sensors, the use of meta-heuristic methods in clustering and routing improves network performance and increases the wireless sensor network's lifetime.

Keywords: Clustering-based Routing; Hybrid Algorithms; Energy Efficiency; Gateway Zone.

1- Introduction

Wireless sensor networks are of the growing technologies with various and broad applications in different fields such as industry, agriculture, military, etc[1]. A wireless sensor network is made of many sensor nodes distributed in the desired environment. In many wireless sensor networks applications, sensor nodes are distributed in a random manner and unplanned in the environment [2]. The sensors are in charge of sensing, collecting, and processing their surroundings' data and sending them to the center or the sink. Because of the limited energy of sensors, minimizing the energy consumption of sensor nodes is one of the main challenges in these networks. The application of clustering techniques is one of the most effective methods in reducing the energy consumption of sensor nodes [3],[4]. The sensor nodes send the collected data to the cluster head node by employing the clustering approach; sending data to the sink is the task of the cluster heads. In general, the applied routing protocols are divided into two groups: flat and hierarchical. In flat protocols, the cluster heads transfer the data to the sink directly and in a single-step manner; this process causes head clusters far from the sink to consume a great deal of energy. However, in hierarchical protocols, the cluster heads send the data hierarchically, rather than directly, to other cluster headers until the data is sent to the sink. In this state, routing between the cluster headers for sending data to the sink is a challenging issue that led to creative and new methods [5]. In general, hierarchical protocols perform better than flat protocols to balance power consumption and extend the network's lifetime. In wireless sensor networks, there are three types of nodes: cluster head, member node, and sink node, each of which is in charge of a function in the wireless sensor network [6]. Member nodes sense environment data and transmit them to the cluster head through multiple time division multiple access (TDMA). Cluster heads receive data from

[1] Optimal clustering-based routing protocol

* Corresponding Author

member nodes and send the collected data to the sink via single-hob or multi-hob. The selection of cluster headers can be performed separately by sensors or sink or maybe pre-implemented by the wireless network designer. In the present paper, the selection of cluster headers has been made by the sink because of enough energy and the capability to make multidimensional calculations [7],[8]. In the present study, the authors present a new innovative algorithm for the hierarchical optimal cooperative routing protocol (OCRP) that performs clustering of sensor nodes and the routing process between cluster heads in each round. The remaining structure of this article's contents is as follows: Section 2 is related to the previous works, the proposed protocol is discussed in Section 3, and results and conclusion will be addressed in Section 4 and Section 5, respectively.

2- Related Works

Due to the expansion of the application of sensors and wireless sensor networks in recent years and the limited and non-rechargeable energy of sensors, increasing the efficiency and lifetime of wireless sensor networks is one of the most important challenges. Hence, various protocols have been presented to solve these basic challenges. Three of these proposed algorithms will be evaluated in the following.

2-1- Application Specific Low Power Routing Protocol (ASLPR)

The ASLPR protocol [9] applies various parameters, such as distance from the base station, the distance between the cluster heads and the sensor node, and remaining energy, for selecting the cluster head. For this purpose, each node first picks a random number between 0 and 1. In the case that the random number assigned to a node is less than T_{ASLPR} in Eq. (1), this node is chosen as the cluster head.

$$T_{ASLPR} = \begin{cases} Z(n) & \text{if } E(n) \geq t_1 \times \frac{1}{N}\sum_{i=1}^{N} E(i) \\ 0 & \text{if } E(n) < t_1 \times \frac{1}{N}\sum_{i=1}^{N} E(i) \end{cases} \quad (1)$$

$$Z(n) = \alpha_1 T_1(n) + \alpha_2 T_2(n) + \alpha_3 T_3(n) + \alpha_4 T_4(n) \quad (2)$$

Where N implies the total number of live nodes in the current frequency, and $E(n)$ equals the remaining energy of node n. In Eq. (2), $T_1(n)$ is referred to the partial energy threshold of the nodes, and α_1 denotes the weight of this partial threshold. Moreover, $T_2(n)$ is the partial threshold for the distance between the base station and the nodes, as well as α_2 is referred to as the weight of this partial threshold. Meanwhile, $T_3(n)$ is the partial threshold for the distance between the cluster head and the node, and α_3 implies the weight of this partial threshold. The partial threshold $T_4(n)$ implies the number of frequencies in which a node is the cluster head, and α_4 denotes the weight of this partial threshold.

2-2- Clustering Routing Protocol for Dynamic Network (CRPD)

CRPD [10] provides a clustering-based routing protocol for dynamic networks and consists of 4 steps: 1) Neighbor discovery, 2) cluster head selection and cluster formation, 3) path construction and data collection, and 4) re-clustering and rerouting. This algorithm's basic idea is to select a sensor node with more energy and a larger degree as a cluster head in charge of collecting data transmission in each round.

In this algorithm, four principles are followed to select a cluster head and create a cluster as follows:

1. Sensor nodes with the highest degree (highest number of neighbors) are selected.
2. The remaining energy of the sensor nodes (Er) must be higher than the threshold energy Ethreshold (Ethreshold= 0.4 E0).
3. In the case that the Er of the node with the highest degree is not higher compared to the Ethreshold, another node with the highest degree among the neighbors and the Er higher than Ethreshold is selected as the cluster head.
4. The cluster heads cannot be adjacent. Also, cluster head selection relies on a couple of factors: remaining energy (Er) and the degree of sensor nodes.

According to the formed cluster, every single sensor node transfers the data to the cluster head, and since the nodes are aware of their and sink's positions, the cluster heads select a node closer to the sink to select the next step in sending data from their neighbors to the sink.

2-3- Cuckoo Optimization Algorithm- based Routing Protocol (COARP)

In COARP [11], measurements are made to determine cluster heads in a central control system named sink. The network model is a single-step model in which the cluster heads directly connect with the sink. In each round, the sink is aware of the network nodes' energy level and position, and each node senses and collects surrounding data and then processes and transfers the data to the cluster head in a data packet. The COARP clustering method consists of steps: 1) Setup, which includes determining the cluster head and creating the cluster, 2) Registration, which includes generating the schedule, and transferring the data. In COARP, cluster heads are precisely selected by the Cuckoo algorithm in the sink. Afterward, the process of creating the clusters and the registration step is performed. Each cluster head receives data related to all member nodes in its cluster and then transfers the received data in one step to the central station in the form of a packet.

3- Proposed Protocol

The present article's objective is the reduction of the energy consumption of sensor nodes through clustering and routing in multistage communication. The proposed protocol in this paper consists of four steps: 1) determining the gateway zone, 2) selecting the cluster head, 3) clustering and 4) routing to send the collected data. The general algorithm of the proposed method is as follows:

OCRP Algorithm

```
Divide area to sensing and gateway
Select nodes in sensing area for clustering;
Clusterheads=TLBO Algorithm;
For i=1: number of nodes
    If node_i is in sensing area && node_i is normal node;
        Node_i joins to nearest clusterhead;
    End_IF
End_For
Route=Tabu Search Algorithm;
For i=clusterheads
    CH_i joins to route;
End_For
Route connect to nearest gateway node;
For i=gateway nodes
    If node_i is awake
        Node_i send data to Base Station;
    End_If
End_For
```

3-1- Gateway Zone

The sensor nodes are specified in the distributed environment and the sink’s position in a random manner. Then, 20% of the environment is determined, which is closer to the sink as the gateway zone. The sensor nodes placed in this section are considered as relay nodes, the role of which is to receive data from the cluster headers and send them directly to the sink node.

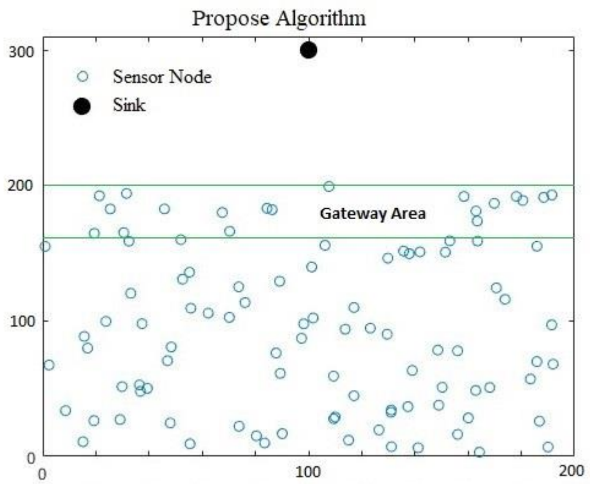


Fig. 1. Distribution of sensor nodes and determining the gateway zone

In this paper, first, a zone is considered as a gateway; the sensors within this zone are responsible for receiving data from the cluster headers and sending them to the sink. The purpose of this paper is the reduction of the energy consumption of sensors at the time of sending and inserting them in a specific cluster in multi-step communications. In this section, the authors will describe the proposed protocol in detail, which includes the following subsections: pseudocode of the proposed protocol, distribution of nodes, selection of cluster heads, cluster headers membership, and routing.

3-2- Selection of Cluster Heads and Clustering

After determining the gateway zone, the sensors outside the gateway zone should be clustered. In order to specify the headers from the existing sensor nodes, the method of probabilistic distribution in TLBO is applied. The TLBO algorithm is a population-based optimization and also a swarm intelligence algorithm inspired by the teaching-learning process in a classroom.

3-2-1- TLBO Algorithm

The TLBO algorithm consists of four phases, as follows:

1) **Initialization:** Creating the initial population in optimization algorithms is essential for obtaining the solution to a specified problem and is useful for optimal problem solving.
2) **Teaching phase:** In this section, the best learners are often picked as teachers; they transfer their knowledge to the learners to improve their knowledge level.
3) **Learning *phase***: In this stage, learners communicate with each other for increasing their knowledge level.
4) **Reviewing after class:** Some learners enhance their effectiveness through reviewing the knowledge.

Learner initialization: Each learner possesses N decision variables where N implies the number of live sensors in that round. A random number in the interval [0,1] is assigned to each variable ($X_i = rand(0,1)$).

Teaching phase: Throughout this phase, the best learner in the population is considered as a teacher. Although, in the case that the best learner falls into the trap of local optimization, it is probable that other learners to not improve effectively. In the meantime, the use of the average of the total population is illogical, which may lead to a gradual evolution of the population. Accordingly, the average of the top learners, which includes the top six learners in the population, is applied instead of the average of the total population. In the present article, the top learners are selected by truncation to build the Gaussian model.

$$N(x_j, \mu_j, \sigma_j) = \frac{1}{\sqrt{2\pi\sigma_j}} \exp(\frac{1}{2}\left(\frac{x_j-\mu_j}{\sigma_j}\right)^2) \qquad (3)$$

Two principle parameters are available for the high Gaussian distribution: standard deviation and mean. These parameters significantly affect the search performance. Due to the fact that the teacher must update his knowledge in a timely manner, a dynamic update mechanism is used for adjusting the Gaussian distribution. Standard deviation determines the search accuracy, and the mean controls the search direction. If the mean fails to reach the overall optimal region, the standard deviation is reduced to its minimum value, and the search will be slowed down. The standard deviation and mean are obtained from Eqs. (4) and (5).

$$\mu_j = (1-\alpha).sup\mu_j + \alpha.(x_{best1,j} + x_{best2,j} - x_{worst,j}) \qquad (4)$$

$$\sigma_j = (1-\alpha).sup\sigma_j + \alpha.who\sigma_j \qquad (5)$$

Where supσ_j and supμ_j imply the standard deviation and mean of better and more effective solutions, respectively, $who\sigma_j$ implies the average of the total population. x_{best1}, x_{best2}, x_{worst} are the best, second best, and worst learners in the population. The sign α implies the learning rate applied for controlling the update rate of Gaussian distribution α϶ (0,1).

For the purpose of avoiding the falling in the trap of local optimization, the teacher uses crossover learning for transferring knowledge to the learner; in other words, if r_i <0.5 for each dimension of a new learner, the current dimension is developed using the Gaussian distribution, otherwise equals the related dimension of the former learner. $r_i \epsilon (0,1)$ (0,1) is a random number that follows the uniform distribution. According to the mentioned scheme, the teaching phase is summarized as follows:

Teaching Phase OCRP Algorithm

```
Initialize learners;
Evaluate learners;
M=6;
Betha=0.1;
While (stopping condition is not met)
Select M superior learners;
SupMu=mean of superior learners;
SupSigma=standard deviation of superior learners;
Mean=mean of all population;
Xbest1=best learner;
Xbest2=second best learner;
Xworst=worst learner;
For i=all learners
    For j=all dimension
        If rand<0.5
            Mui=(1-Alpha) *SupMu+Alpha*(Xbest1+Xbest2-Xworst);
            Sigmai=(1-Alpha)*SupMu+Alpha+Mean;
            Xnewi(j)= sample from the Gaussian distribution;
        Else
            Xnewi(j)=Xi(j);
        End_If
    End_For
    Evaluate new learners;
    If new learner is better than existing
        Xi=Xnewi;
    End_If
End_For
```

Learning phase: In this phase, the learner should be able to discuss not only with classmates but also with the best learner and improve their knowledge according to their current information. This phase consists of two parts: local learning and permutation-based crossover learning, the details of which are presented as follows:

In local learning, local direction and data are integrated into the new learner. According to Eqs. (6) and (7), T_{xnew} temporary learners are developed by a linear combination of x_{rnd}, x_{best}, and a random vector; x_{rnd} implies a random learner picked from a population that is dissimilar to x_i. r_1, r_2, and r_3 are random numbers in the interval [0,1]. The coefficient β is a randomized parameter equal to β=0.1. x_i and x_{rnd} determine the direction of learning, and $\beta.(r_3 - 0.5)$ specifies the local search domain.

$$\mathrm{Tx}_{\mathrm{new}} = \mathrm{x_i} + \mathrm{r_1}.\left(\mathrm{x_i} - \mathrm{x_{rnp}}\right) + \mathrm{r_2}.(\mathrm{x_{best}} - \mathrm{x_i}) + \beta.(\mathrm{r_3} - 0.5), \ \ \text{if } \mathrm{f(x_i) < f(x_{rnp})} \qquad (6)$$

$$\mathrm{Tx}_{\mathrm{new}} = \mathrm{x_i} + \mathrm{r_1}.\left(\mathrm{x_{rnp}} - \mathrm{x_i}\right) + \mathrm{r_2}.(\mathrm{x_{best}} - \mathrm{x_i}) + \beta.(\mathrm{r_3} - 0.5), \ \ \text{if } \mathrm{f(x_i) \geq f(x_{rnp})} \qquad (7)$$

Learners only communicate with their classmates during the learning phase. Obviously, learners disregard the knowledge exchange that would lead to the loss of historical data. Accordingly, the differential evolution's double crossover is applied to save some historical data in each learner. In Eq. (8), the crossover function is described where C_r denotes the crossover rate in the interval [0,1]. $j_{rnd} \epsilon \{1,2,\dots,n\}$ is a randomly selected index which ensures that x_{new} receives at least one dimension from Tx_{new}.

$$x_{new,j} = \begin{cases} Tx_{new,j} & \text{if}(r_i \leq Cr \text{ or } j = j_{rnd}) \\ x_{old,j} & \text{otherwise} \end{cases} \quad (8)$$

In the above phase, learners exchange knowledge only between continuous vectors. A two-point (TP) crossover operator is randomly picked from two previous learners x_{r1}, x_{r2}, and a population that is different from the current learner. Afterward, a new pair of learners $x1_{new}$ and $x2_{new}$ are created after the crossover operation. If the better learner from x1new and x2new shows better performance than the current learner, the better learner will be replaced by the current learner.

Learning phase OCRP Algorithm

```
For i=all learners
    Randomly select another learner which is different from i (Xk);
    Generate a temporary learner;
    If Xi is better than Xk
        TXnew=Xi+rand*(Xi-Xk) +rand*(Xbest-Xi) +Betha*(rand-0.5);
    Else
        TXnew=Xi+rand*(Xk-Xi) +rand*(Xbest-Xi) +Betha*(rand-0.5);
    End_If
    For j=all dimension
        If rand<=Cr OR j=Jrnd
            Xnewi(j)=Txnew(j);
        Else
            Xnewi(j)=Xi(j);
        End_If
    End_For
    If new learner is better than existing
        Xi=Xnewi;
    End_If
End_For
For i=all new learners
    Randomly select two learners x1 and x2 from the population that are different from Xi;
    x1new=TP (x1, Xi);
    x2new=Tp (x2, Xi);
    If x1new is better than x2new
        If x1new is better than Xi
            Xi=x1new;
        End_If
    Else
        If x2new is better than Xi
            Xi=x2new;
        End_If
    End_If
End_For
```

3-3- Routing

The cluster heads receive and collect data from the member sensor nodes of their cluster; then, the data must be sent to the sink. For this purpose, hierarchical routing should be applied so that the cluster heads send data to the sink in a multistage manner through creating an optimal path by the combination of TLBO and TS algorithms. According to the quantity of cluster heads in the network environment, initially, the TLBO algorithm is applied, and a response is obtained, then the response is optimized by the TS algorithm, and the cluster heads determine the optimal path for sending data. In this method, the authors first obtain two populations with 50 members, calculate, and sort the costs of each learner, and then select the least costly learners based on the number of learning members in the population. Each member of the population has a Cost and Position variables equal to the number of cluster heads minus one (N_{Ch}-1). The first population is created randomly, and the second population is obtained using Eq. (9).

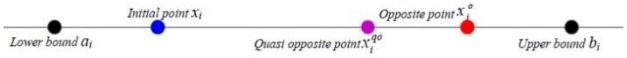


Fig. 2. Opposition-based learning and quasi-oppositional learning

$$x_i^o = a_i + b_i - x_i \quad (9)$$

$$x_i^{qo} = \frac{a_i+b_i}{2} + \text{rand}.\left(x_i^o - \frac{a_i+b_i}{2}\right) \quad (10)$$

Teaching phase: The update mechanism during the teaching phase is described as follows:

$$\text{new}X_i = X_i + \text{rand}.(\text{Teacher} - \text{TF}.\text{Meam}) \quad (11)$$

$$\text{Mean} = \frac{1}{NP}\sum_{i-1}^{NP} X_i \quad (12)$$

Where, $newX_i$ implies the new position of the learner, X_i denotes the nth learner, *Teacher* implies the teacher with the best fitting, NP is referred to the quantity of learners in the population, and TF is the teaching factor which determines the size of the average to be altered. *rand* denotes a random vector, and its element is a random number in the interval [0,1].

Learning phase: The update mechanism for the *i*th learner during the learning phase is expressed as follows:

$$\text{new}X_i = \begin{cases} X_i + \text{rand}.(X_i - X_k) & \text{if } f(X_i) < f(X_k) \\ X_i + \text{rand}.(X_i - X_k) & \text{otherwise} \end{cases} \quad (13)$$

Where *newXi* denotes the new position of the *i*th learner, X_k is the randomly selected learner from the class. The learner fitting values of X_i and X_k are represented by $f(X_i)$, and $f(X_k)$, respectively, and *rand* is a random vector in the interval [0,1].

After completing the TLBO phases in each iteration, the population with the best cost is selected and given to the TS algorithm. This algorithm is implemented to the specified number of iterations and optimizes the desired response. In order to calculate the cost, the Prüfer algorithm is applied to form a tree between the nodes of the cluster head, and its cost is calculated.

The TS algorithm receives a solution, and actions, including Swap, Reversion, and Insertion, are applied to the solution variables. According to the considered actions, all states of actions related to this operation is created in a list called Action List. These actions are applied to the obtained solution, and the cost and position are updated for each action. If the cost is lower, it replaces the best solution, and the action is placed on the Tabu List until a certain number of rounds are performed. The desired number of actions is calculated using Eq. (14) as follows:

$$\mathrm{N_{Action}} = \mathrm{N_{Swap}} + \mathrm{N_{Reversion}} + \mathrm{N_{Insertion}} \tag{14}$$

$$\mathrm{N_{Swap}} = \mathrm{n} \times {}^{(\mathrm{n}-1)}/_{2}$$

$$\mathrm{N_{Reversion}} = \mathrm{n} \times {}^{(\mathrm{n}-1)}/_{2}$$

$\mathrm{N_{Insertion}} = \mathrm{n} \times \mathrm{n}$ (n = number of position variables)
$\mathrm{Tabu\ List} = \mathrm{Round}(0.5 \times \mathrm{N_{Action}})$

In order to calculate the cost of each solution, first, it is converted to the corresponding tree using the Prüfer algorithm, then routing is performed according to the obtained tree, and the cost is calculated from Eq. (15) where E_1 denotes the grid energy before application and E_2 implies the energy calculated after applying the routing operation.

$$\mathrm{Cost} = \mathrm{E_1} - \mathrm{E_2} \tag{15}$$

This process continues until obtaining the best solution. Finally, the solution is given to the Prüfer algorithm with an optimal tree as its output according to which the routing is performed. If the maximum iteration of the algorithms in routing is considered as M, the number of iterations of the TS algorithm is equal to M × M since TS is entirely implemented for each iteration of the TLBO algorithm.

3-4- Network Operations and Calculation of Energy Consumption

In the proposed algorithm, network operations are divided into setup and stability phases. Each node's energy consumption in each round is calculated by evaluating what happened in both stages.

3-4-1- Setup Phase

Sink employs the k_{CP} control packet to communicate with sensor nodes. These control packets contain short messages that request position, energy level, and ID from all sensor nodes. For the purpose of receiving control packets from the sink, as in Eq. (16), the energy $E_{Rx}(k_{\mathrm{CP}})$ is consumed. Moreover, all nodes also use the energy $\mathrm{E_{Tx}}(k_{\mathrm{CP}}, \mathrm{d})$ to transfer control packets, comprising of data around their positions, energy levels, and IDs, to the sink.

$$\mathrm{E_{Rx}(k) = kEelect} \tag{16}$$

$$\mathrm{E_{Tx}(k,d)} = \begin{cases} \mathrm{KE_{elect}} + \varepsilon_{\mathrm{mp}}\mathrm{Kd^4}, & \mathrm{if\ d > d_0} \\ \mathrm{KE_{elect}} + \varepsilon_{\mathrm{fs}}\mathrm{Kd^2}, & \mathrm{if\ d < d_0} \end{cases} \tag{17}$$

Where $d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$ depends on the threshold distance, amplifier energy, ε_{mp} or ε_{fs} , receiver distance, and the allowable rate of bit error. According to the proposed algorithm, the sink processes the control packets and specifies which nodes become the cluster head, and each node becomes the member of which cluster head. Additionally, all nodes also use $E_{Rx}(k_{\mathrm{CP}})$ energy for receiving their status data from the sink (whether members or CH). The consumed energy of all cluster heads to send TDMA schedules to members is presented in Eq. (18) as follows:

$$\mathrm{E_{Tx(ch_i)}(K_{CP}, d_{i-tomem})} = \\ \textstyle\sum_{\mathrm{i=1}} \mathrm{ch_i} * \begin{cases} \mathrm{K_{CP}E_{elect}} + \varepsilon_{\mathrm{mp}}\mathrm{K_{CP}d^4_{i-tomem}}, & \mathrm{if\ d < d_0} \\ \mathrm{K_{CP}E_{elect}} + \varepsilon_{\mathrm{fs}}\mathrm{K_{CP}d^2_{i-tomem}}, & \mathrm{if\ d > d_0} \end{cases} \tag{18}$$

The members consume energy for receiving TDMA schedules from the cluster head, which is calculated using Eq. (17).

3-4-2- Stability Phase

In a steady-state, active nodes send k-bit data to their cluster head according to the TDMA schedule received from the sink. The cluster head is all the time ready to receive this sensed data from its members and processes and collects all the data received from its members before transferring to the sink. The consumed energy by the transmitter sensor of the cluster head, $\mathrm{E_{DA}}$, is calculated by Eq. (17).

$$\mathrm{E_{DA(m_i+1)}(k) = KE_{DA} * (\textstyle\sum_{i=1} m_i + 1)} \tag{19}$$

The dissipated energy for transferring sensed data to the cluster head is calculated using Eq. (20) as follows:

$$\mathrm{E_{Rx(m_i)}(k) = \textstyle\sum_{i=1} m_i KE_{elec}} \tag{20}$$

Where m_i implies the member nodes in the series $i = 1, 2, 3, \ldots, n - L$. The signs L and n are the total number of cluster heads and sensor nodes, respectively. Consumed energy by the cluster head for collecting the sensed data from itself and members is calculated by Eq. (19) as follows:

Figure (3) demonstrates a simulation view of the proposed protocol when being simulated.

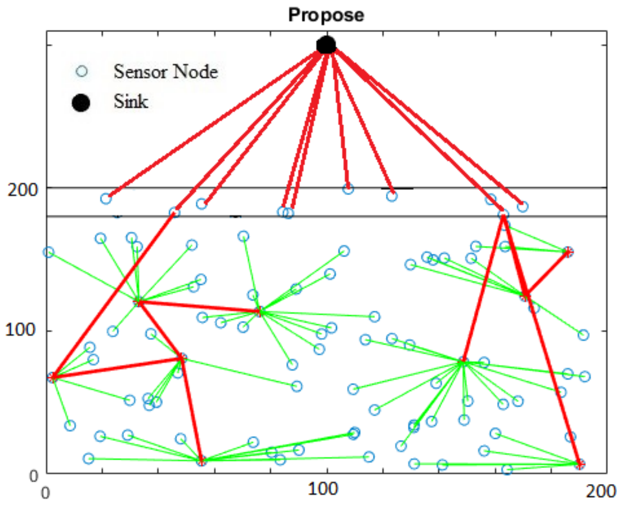


Fig. 3. A simulation view of the proposed method in MATLAB software

4- Results and Discussions

All algorithm simulations have been performed in MATLAB R2019b. For proving the strength of the proposed protocol in various scenarios, it is compared with known protocols such as ALSPR, CRPD, and COARP in terms of First Node Dead (FND), Half Node Dead (HND), Last Node Dead (LND), and the total number of data packets received in the sink from the beginning of the simulation to end of the network lifetime.

Assumptions

In the proposed protocol, the important assumptions of the network and radio models are given as the following:

- The sink is a rich source and is located in a fixed position.
- All sensors are fixed after distribution to the environment, and the energy of all sensors is identical at the beginning of the process.
- All sensors possess global positioning systems or other positioning devices connected to them.
- The desired communication channel is assumed to be symmetric.

Table 1. Adjusting the parameters of the TLBO algorithm

Parameter	Value
Population or Learner	50
Number of iterations	100
Number of Variables	length (Alive Nodes)
Variables Lower Bound	VarMin= 0
Variables Upper Bound	VarMax=1

Table 2. Adjusting the parameters of the TS algorithm

Parameter	**Value**
Population or Solution	1
Number of iterations	100
Number of Variables	Nch-1 (Nch= Number of Cluster Head)
Variables Lower Bound	VarMin= 0
Variables Upper Bound	VarMax=1
N_{Action}	$N_{Swap}+N_{Reversion}+N_{Insertion}$
$N_{Swap} = N_{Reversion}$	N × (N-1)/2
$N_{Insertion}$	N × N (N=Number of position variables)

Table 3. Simulation parameters

Parameter	**Value**
Initial energy of the nodes	1j
$□_{fs}$	10 (pj/bit/m^2)
$□_{mp}$	0.0013 (pj/bit/m^4)
E_{elec}	50 (nJ/bit)
E_{da}	5 (nJ/bit)
Data packet size	4100 (bit)

Table 4. Used scenarios

Number	Number of sensors	Network size	Sink location
1	200	200 m × 200 m	(100 m, 250 m)
2	200	400 m × 400 m	(200 m, 450 m)
3	200	600 m × 600 m	(300 m, 650 m)
4	300	200 m × 200 m	(100 m, 250 m)
5	300	400 m × 400 m	(200 m, 450 m)
6	300	600 m × 600 m	(300 m, 650 m)

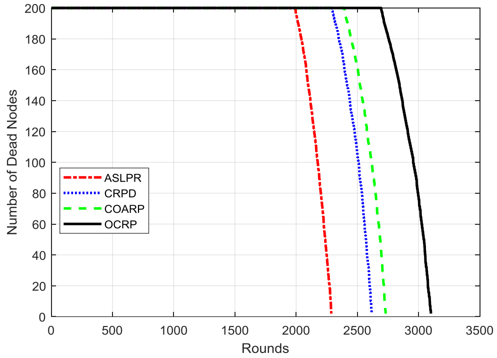


Fig. 4. Number of dead nodes in each round in the first scenario.

Based on the results obtained in Figure (4) in the first scenario, FND, HND, and LND in the proposed method have been increased and showed better performance compared to the ALSPR, CRPD, and COARP by 30%, 17%, and 12%, respectively, which indicates that in the proposed method, the energy of the sensors in each round has been consumed less than other methods. Figure (5) demonstrates the comparison of the network's lifetime, which shows that the proposed method performed better than other methods in consuming the total network energy and spent on average less energy in each round for clustering and sending data.

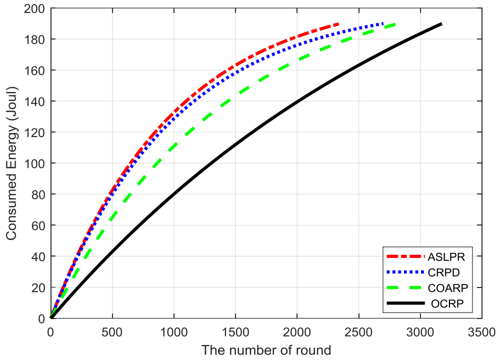


Fig. 5. Network energy consumption in each round in the first scenario.

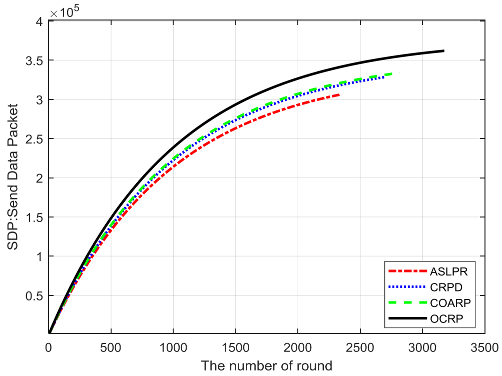


Fig. 6. Packets sent to the sink in each round in the first scenario.

Figure (6) indicates the packets sent to the sink in each round, which shows the increase in the number of packets to be sent to the sink in each round by the proposed method in comparison with other approaches.

In the first, second, and third scenarios, the number of sensors distributed in the environment is constant; however, the size of the network environment and the sink's position is changed to evaluate the performance of the proposed method compared to other techniques. The results obtained from the second and third scenarios, as the first scenario, indicate that the proposed method has led to an increase in FND, HND, and LND compared to other methods. Moreover, the network's lifetime and the number of packets to be sent to the sink in the proposed method have been increased compared to other methods

Table 5. Comparison of FND, HND, and LND of the proposed method with other methods with 200 node sensors.

	Environment= 200 × 200 Sink= 100 × 250			Environment= 400 × 400 Sink= 200 × 450			Environment= 600 × 600 Sink= 300 × 650		
	FND	HND	LND	FND	HND	LND	FND	HND	LND
ASLPR	1992	2188	2349	397	630	1028	44	131	154
CRPD	2290	2516	2701	456	721	1180	49	148	175
COARP	2390	2625	2818	476	750	1222	52	155	182
My Propose (OCRP)	2691	2953	3175	538	845	1367	61	176	205

As can be seen in Table 5, our proposed method performs better in the death of the first node, HND and LND, in different scenarios than the ASLPR, CRPD, and COARP methods. Increasing the simulation environment and the number of sensor nodes is constant, causing the nodes to spend more energy to communicate with each other, and as a result, this increases energy consumption and reduces the life of the network, the results shown in Table 5 testify to This is a claim.

In the fourth, fifth, and sixth scenarios, the number of nodes is increased, and by increasing the number of sensor nodes, the same simulations are carried out. The network's lifetime and the number of packets to be sent to the sink are increased with an increase in the number of sensor nodes within the network. According to Figure (7), FND, HND, and LND in the proposed method are higher than other methods due to the increase in the number of sensor nodes in the network environment in the fourth scenario. Figures (8) and (9) show the better efficiency of the proposed method with regard to the number of packets to be sent to the sink and the network's lifetime.

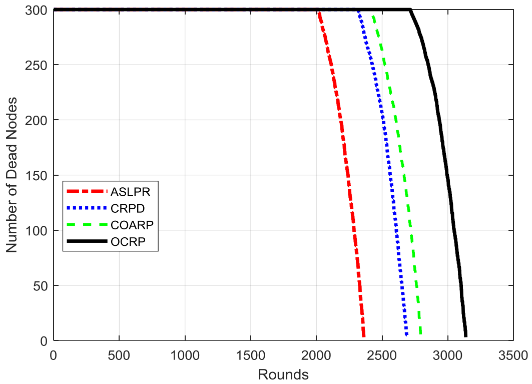


Fig. 7. Number of live nodes in each round in the fourth scenario.

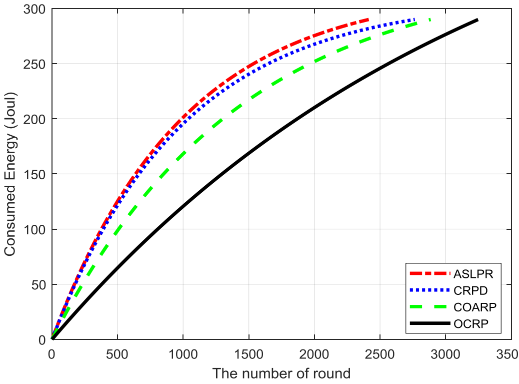


Fig. 8. Network energy consumption in each round in the fourth scenario.

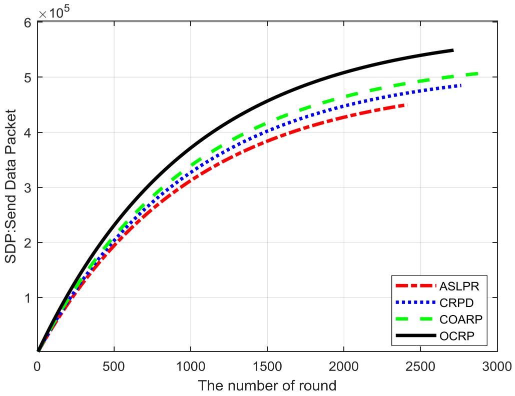


Fig. 9. The packets sent to the sink in each round in the fourth scenario.

As the simulation environment increases, the sensors must consume more energy to communicate with each other because more energy is required for transferring data as well as routing with the increase in the distance of sensor nodes. According to Table (6), the obtained results for FND, HND, and LND of the proposed method compared to other approaches are shown.

Table 6. Comparison of FND, HND, and LND of the proposed method with other methods with 300 node sensors.

	Environment= 200 × 200 Sink= 100 × 250	FND	HND	LND	Environment= 400 × 400 Sink= 200 × 450	FND	HND	LND	Environment= 600 × 600 Sink= 300 × 650	FND	HND	LND
ASLPR		2014	2210	2414		423	712	1105		48	137	158
CRPD		2315	2540	2765		486	818	1270		55	157	180
COARP		2409	2645	2886		507	854	1326		57	164	186
My Propose (OCRP)		2715	2975	3247		571	961	1491		66	185	210

By increasing the number of sensor nodes to 300 and simulating in 3 different environments, the proposed method has increased the network life compared to other methods. Therefore, according to Tables 5 and 6, we can conclude that in different cases, increasing the number of sensor nodes and changing the size of the simulation environment, the performance of the proposed method is better than other methods and has a longer lifetime.

5- Conclusion

In recent years, in wireless sensor networks, different protocols such as ALSPR, CRPD, and COARP have been proposed; the main purpose of each is to increase the network’s lifetime. The use of clustering and hierarchical routing leads to the reduction of the sensors' energy consumption and, ultimately, the increase in the wireless sensor network’s lifetime. In the present article, the selection of the optimal cluster heads from the existing nodes is performed, and the sensors are clustered using the novel approach called Gaussian TLBO; then the multistage routing and the TS_ TLBO hybrid algorithm are applied to transfer the data collected by the cluster heads so that the data is sent to the gateway nodes and finally to the sink.
According to the simulation results, the proposed algorithm reduces sensor nodes' energy consumption because of the proper selection of cluster heads from the available sensor nodes. Also, due to hierarchical routing and the use of gateway nodes, the energy of cluster heads needed to transfer data to the sink is decreased, which leads to an increase in the network's lifetime. The proposed OCRP algorithm reduces the energy consumption compared to the ALSPR, CPRD, and COARP algorithms by approximately 35%, 17%, and 12%, respectively. As future work, the following suggestions can be made to improve and expand the proposed method:

1) The use of the sleep/wakeup technique for gateway nodes to reduce their energy consumption.
2) Making the sink a stimulant to collect data

3) The use of reinforcement learning algorithms for selecting the cluster head.

References

[1] N. R. Roy and P. Chandra, “Energy dissipation model for wireless sensor networks: a survey,” Int. J. Inf. Technol., vol. 12, no. 4, pp. 1343–1353, 2020.

[2] M. Sedighimanesh* and H. Z. H. and A. Sedighimanesh, “Routing Algorithm based on Clustering for Increasing the Lifetime of Sensor Networks by Using Meta-Heuristic Bee Algorithms,” International Journal of Sensors, Wireless Communications and Control, vol. 10, no. 1. pp. 25–36, 2020.

[3] A. Belfkih, C. Duvallet, and B. Sadeg, “A survey on wireless sensor network databases,” Wirel. Networks, vol. 25, no. 8, pp. 4921–4946, 2019.

[4] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, “Clustering objectives in wireless sensor networks: A survey and research direction analysis,” Comput. Networks, vol. 180, p. 107376, 2020.

[5] V. Parashar, B. Mishra, and G. S. Tomar, “Energy Aware Communication in Wireless Sensor Network: A Survey,” Mater. Today Proc., vol. 29, pp. 512–523, 2020.

[6] D. K. Sah and T. Amgoth, “Renewable energy harvesting schemes in wireless sensor networks: A Survey,” Inf. Fusion, vol. 63, pp. 223–247, 2020.

[7] M. Sedighimanesh* and H. Z. and A. Sedighimanesh, “Presenting the Hybrid Algorithm of Honeybee - Harmony in Clustering and Routing of Wireless Sensor Networks,” International Journal of Sensors, Wireless Communications and Control, vol. 9, no. 3. pp. 357–371, 2019.

[8] A. Kochhar, P. Kaur, P. Singh, and S. Sharma, “Protocols for wireless sensor networks: A survey,” Journal of Telecommunications and Information Technology. 2018.

[9] M. Shokouhifar and A. Jalali, “A new evolutionary based application specific routing protocol for clustered wireless sensor networks,” AEU - Int. J. Electron. Commun., vol. 69, no. 1, pp. 432–441, Jan. 2015.

[10] S. Wang, J. Yu, M. Atiquzzaman, H. Chen, and L. Ni, “CRPD: a novel clustering routing protocol for dynamic wireless sensor networks,” Pers. Ubiquitous Comput., vol. 22, no. 3, pp. 545–559, 2018.

[11] M. Khabiri and A. Ghaffari, “Energy-Aware Clustering-Based Routing in Wireless Sensor Networks Using Cuckoo Optimization Algorithm,” Wirel. Pers. Commun., vol. 98, no. 3, pp. 2473–2495, 2018.

Ali Sedighimanesh received the B.S. degree in Information technology from Islamic Azad University, Parand Branch, Iran in 2009, and M.S. degree in Information technology from Islamic Azad University, Qazvin Branch, Iran, in 2013. Currently, he is Ph.D. Candidate in Islamic Azad University, Tehran Branch, Iran. His research interests include wireless sensor networks, IoT (internet of things), Blockchain, Web mining, and Data mining.

Hessam Zandhessami is an assistant professor of Information technology at Azad University, Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran. His research is focused on, IOT (internet of things), Web mining and Methodologies in Software Engineering, Management.

Mahmood Alborzi is a full-time Assistant-Professor of Information technology at Azad University, Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran. He received his Ph.D. degree PhD in Artificial Neural Networks Brunel University, UK, England in ۱۹۹٦. He is conducting research activities in the areas of Image and Data Analysis Software Systems, Artificial Neural Networks, data mining.

Khayyatian mohammadsadegh is a full-time Assistant-Professor of technology management at Shahid Beheshti University, Tehran, Iran. He received his Ph.D. degree PhD in technology management Allameh Tabatabaei University. He is conducting research activities in the areas of Image and Data Analysis Software Systems, Artificial Neural Networks, data mining, technology management.

Improvement of Firefly Algorithm using Particle Swarm Optimization and Gravitational Search Algorithm

Mahdi Tourani*
Technical faculty of Ferdows, University of Birjand, Iran
tourani.mahdi@birjand.ac.ir

Received: 11/Oct/2020 Revised: 27/Mar/2021 Accepted: 11/Apr/2021

Abstract

Evolutionary algorithms are among the most powerful algorithms for optimization, Firefly algorithm (FA) is one of them that inspired by nature. It is an easily implementable, robust, simple and flexible technique. On the other hand, Integration of this algorithm with other algorithms, can be improved the performance of FA. Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA) are suitable and effective for integration with FA. Some method and operation in GSA and PSO can help to FA for fast and smart searching. In one version of the Gravitational Search Algorithm (GSA), selecting the K-best particles with bigger mass, and examining its effect on other masses has a great help for achieving the faster and more accurate in optimal answer. As well as, in Particle Swarm Optimization (PSO), the candidate answers for solving optimization problem, are guided by local best position and global best position to achieving optimal answer. These operators and their combination with the firefly algorithm (FA) can improve the performance of the search algorithm. This paper intends to provide models for improvement firefly algorithm using GSA and PSO operation. For this purpose, 5 scenarios are defined and then, their models are simulated using MATLAB software. Finally, by reviewing the results, It is shown that the performance of introduced models are better than the standard firefly algorithm.

Keywords:K-best Attractive Firefly; Global and Local Best Position; Gravitational Search Algorithm (GSA); Improved Firefly Algorithm (IFA); Movement in Algorithm; Particle Swarm Optimization.

1- Introduction

Since most real-life problems can be modeled as optimization tasks, many methods and techniques that could tackle such problems were developed. Thus, the optimization became one of the most applicable fields in mathematics and computer science. The difficulty of an optimization problem depends on the mathematical relationships between the objective function, potential constraints, and decision variables. Hard optimization problems can be combinatorial (discrete) or continuous (global optimization), while continuous problems can be further be classified as constrained or unconstrained (bound constrained) [1].

Today, different algorithms have been proposed to solve optimization problems, one of them is Evolutionary Algorithms (EA). These algorithms are inspired by nature and offer different solutions to search in the problem space. EAs are stochastic optimization methods based on the evolution theory. They handle a population of candidate solutions (offspring) that evolves according to the principles of natural selection; that is, using selection, recombination, and mutation processes. During the evolution, individuals compete, and the fittest among them mate for creating the offspring population.

EAs are able to locate the global optimum and are widely used in engineering optimization problems because they may accommodate any ready-to-use evaluation software. However, EAs call for a great number of fitness function evaluations before reaching the global optimum [2].

One of the most popular of EA is the Firefly Algorithm (FA). Fireflies are winged beetles or insects that produce light and blinking at night. The light has no infrared or an ultraviolet frequency which is chemically produced from the lower abdomen is called bioluminescence. They use the flash light especially to attract mates or prey. The flash light also used as a protective warning mechanism to remind the fireflies about the potential predators. Firefly algorithm formulated by Yang [3] is a metaheuristic algorithm that is inspired by the flashing behavior of fireflies and the phenomenon of bioluminescent communication. Reference [3] formulated the firefly algorithm with the following assumptions:

1) A firefly will be attracted to each other regardless of their sex because they are unisexual.

2) Attractiveness is proportional to their brightness whereas the less bright firefly will be attracted to the

* Corresponding Author

brighter firefly. However, the attractiveness decreased when the distance of the two fireflies increased.

3) If the brightness of both fireflies is the same, the fireflies will move randomly [4].

After [3], authors in [5-11] and some other improved Firefly optimization and contributed to the development of this algorithm. Improvement of Firefly Algorithm (FA) in [5] with Asexual Reproduction Optimization Algorithm (ARO), in [6] with Artificial Bee Colony (ABC) and in reference [7] using Pattern Search (PS) are done. There are also many research on the application of the firefly algorithm to various issues [12-15]. Such as these, Engineering optimization problems can be addressed.

The aims of this paper is improvement the firefly algorithm (FA) performance using other algorithms such as Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA).

In the firefly algorithm, the search is based on current information, but in the PSO, algorithm preserves past data such as local best answer and global best answer to used in new candidate answer. Using this PSO feature and combine it with the firefly algorithm, the behavior of improved FA (Introduced in this paper) will be a function of current and past information.

On the other hand, one of the positive feature of the GSA is the use of more effective information in generating new candidate answer. The main purpose of this, is to reduce the running time of the algorithm, but this method can be avoided confusion in choosing the optimal answer due to the high volume of information too. Therefore, using this GSA feature and combine it with the firefly algorithm, the improved FA (that Introduced in this paper), will use only useful information in problem search processing instead of all the information - effective or ineffective -.

These reasons and other advantages, make the PSO and GSA a good candidate to improve the FA. The high speed and the accuracy of the Improved FA, can be of great help in solving the engineering optimization problems such as Smart Grids, Electric Vehicles and etc.

2- Standard Firefly Algorithm

There are three basic rules in the firefly algorithm that are mentioned in the introductory section. It should be noted that the FA is based on these rules. Initially, a random position of fireflies is produced. The brightness of the fireflies should be associated with the objective function of the related problem.

2-1- The Attractiveness of the Firefly

The attractiveness of firefly i on the firefly j is based on the degree of the brightness of the firefly i and the distance r_{ij} between the firefly i and the firefly j [16] as in Eq. (1).

$$I(r) = \frac{I_s}{r^2} \tag{1}$$

It is supposed that there are n fireflies; and x_i corresponds to the solution for firefly i. The brightness of the firefly i, is associated with the objective function $f(x_i)$. The brightness I of a firefly is chosen to reveal its recent position of its fitness value or objective function f(x) as in Eq. (2).

$$I_i = f(\mathrm{x}_i) \tag{2}$$

The less bright (attractive) firefly is attracted and moved to the brighter one. Each firefly has a certain attractiveness value β. However, the attractiveness value β is relative based on the distance between fireflies. The attractiveness function of the firefly is established by Eq. (3).

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{3}$$

Where β_0 is the firefly attractiveness value at r=0 and γ is the media light absorption coefficient [4].

2-2- The Movement Towards Attractive Firefly

It is worth pointing out that the exponent γ_r can be replaced by other functions such as γ_{rm} when m>0. The distance between any two fireflies i and j at x_i and x_j can be Cartesian Distance in Eq. (4).

$$r_{ij} = \left\| x_i - x_j \right\|_2 = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} \tag{4}$$

The firefly i movement is attracted to another more attractive (brighter) firefly j is determined by Eq. (5):

$$x_i^n = x_i^{n-1} + \beta_0 e^{-\gamma r_{ij}^2} \left(x_j^{n-1} - x_i^{n-1} \right) + a e_i^{n-1} \tag{5}$$

Where x_i^n is position of firefly *i* in iteration *n*. In this equation, the second term is due to the attraction, while the third term is randomization with the vector of random variable ε_i being drawn from a Gaussian Distribution and (α Є [0,1]) [5].

The flowchart of the Standard Firefly Algorithm is shown in Fig. 1.

3- Proposed Algorithms

In the standard version of the firefly algorithm, the main role in problem space search is played by the Movement Operator. As this operator is better, the FA is faster and more accurate. In this paper, Gravitational Search Algorithm (GSA) and Particle Swarm Optimization (PSO) are used to improve the performance of Standard FA. These algorithms help to improve Movement Operator in problem space search.

3-1- Improvement of Firefly Algorithm using Gravitational Search Algorithm

The Gravitational Search Algorithm (GSA) is a kind of population based stochastic search algorithm which was first proposed by Rashedi et al. [17], [18].

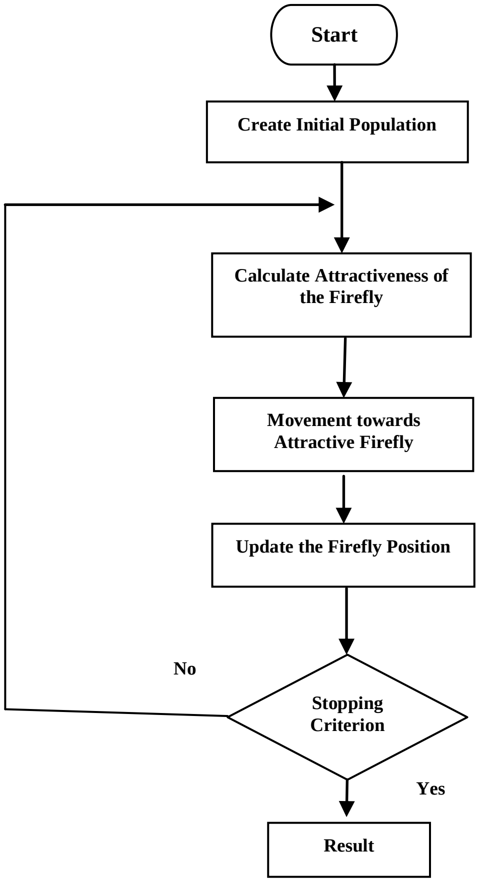

Start
Create Initial Population
Calculate Attractiveness of the Firefly
Movement towards Attractive Firefly
Update the Firefly Position
No
Stopping Criterion
Yes
Result

Figure 1. Flowchart of the Standard Firefly Algorithm

The method is based on Newton's theory. Newton's law states that every particle (mass) attracts another particle by means of some gravitational force. Technically, in GSA, each particle has associated with four attributes: particle position, its inertial mass, active gravitational mass, and passive gravitational mass. The particle's position gives the solution of a problem while fitness function is used to calculate the gravitational and inertial masses [19].

In gravitational search algorithm, the movement plays an important role in searching the problem space. The force exerted on the particles by other masses causes this movement. This force is obtained by Eq. (6-10)

$$F_i^n = \sum_{j=1}^{m} F_{ij}^n \times rand_j \tag{6}$$

$$F_{ij}^n = G^n \frac{M_i^n \times M_j^n}{(\mathrm{r}_{ij}^n)^p + \varepsilon}(x_j^n - x_i^n) \tag{7}$$

$$G^n = G_0 e^{\frac{-\alpha n}{Max(n)}} \tag{8}$$

$$M_i^n = \frac{q_i^n}{\sum\limits_{i=1}^{m} q_i} \tag{9}$$

$$q_i^n = \frac{fit_i^n - worst^n}{best^n - worst^n} \tag{10}$$

where, F_i^n is the force on the *ith* particle in *nth* iteration, $rand_j$, random value, F_{ij}^n, the force on the *ith* particle from *jth* particle in *nth* iteration, G_0, primary gravitational constant, $Max(n)$, maximum of iterations, r_{ij}, the distance between *ith* and *jth* particles, fit_i^n, fitness of *ith* particle, $best^n$ and $worst^n$ are best and worst fitness of *n* iteration.

By creating force on masses, the particles are accelerated and eventually moved.

$$a_i^n = \frac{F_i^n}{M_i^n} \tag{11}$$

$$V_i^{n+1} = rand_i \times V_i^n + a_i^n \tag{12}$$

$$x_i^{n+1} = x_i^n + V_i^{n+1} \tag{13}$$

where, a_i^n is acceleration of the *ith* particle in *nth* iteration, V_i^{n+1}, *ith* particle velocity in *(n+1)th* iteration and x_i^n is position of *ith* particle in *nth* iteration.

In Eq. (6), calculating the effect of all masses on each particle is take a long time and it also can reduce the accuracy of the optimal response, therefore, in one version of the GSA, K-best particles with bigger mass are used to calculate the force exerted on the particles, rather than calculating the effect of all.

$$F_i^n = \sum_{j=1}^{kbest} F_{ij}^n \times rand_j \tag{14}$$

where, *kbest* is *k* particle with bigger fitness in *nth* iteration.

This can improve the performance of the algorithm. So that, for proposed FA in this paper, it is suggested to be used *Kbest* for Movement Operator. That way, each firefly is attracted to K-best more attractive instead of all more attractive fireflies. therefore, The Eq. (5) is modified as the Eq. (15).

$$x_i^n = x_i^{n-1} + \beta_0 e^{-\gamma r^2_{ij\in Kest}} \left(x^{n-1}_{j\in Kbest} - x_i^{n-1} \right) + ae_i^{n-1} \tag{15}$$

where, $j \in Kbest$ Specify just K-best of more attractive fireflies can be effective in Eq. (15).

It is expected to improve the speed and accuracy of the firefly algorithm with this change in movement towards attractive firefly.

3-2- Improvement of Firefly Algorithm using Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm is a member of the wide category of swarm intelligence methods for solving global optimization problems. It was originally proposed by Kennedy as a simulation of social behavior, and it was initially introduced in 1995 as an optimization method [20,21]. PSO is a population based optimization tool, where the system is initialized with a population of random particles and the algorithm searches for optimal by updating generations [22].

As the GSA, in Particle Swarm Optimization (PSO), the movement generates a new candidate response for problem space. For this movement, the particle velocity is calculated by Eq. (16).

$$V_i^{n+1} = c_1\left(x_i^{localbest} - x_i^n\right) + c_2\left(x^{globalbest} - x_i^n\right) + wV_i^n \qquad (16)$$

where, $x_i^{localbest}$ best historical position, $x^{globalbest}$ best position of the entire population, V_i^n *ith* particle velocity in *nth* iteration and c_1 , c_2 and *w* are impact factors for moving to local or global best answer and old particle velocity.

After calculating the velocity, the new position of the particles is obtained by Eq. (17)

$$x_i^{n+1} = x_i^n + V_i^{n+1} \qquad (17)$$

In PSO, candidate answers for each iteration calculate according to the distance of its current position from both its own best historical position and the best position of the entire population or its neighborhood. This operation can be defined as movement towards local and global optimum point. it is the key to achieving best response in PSO and can be used to improve the performance of the Firefly algorithm.

$$x_i^n = x_i^{n-1} + c_1\beta_1\left(x_i^{localbest} - x_i^{n-1}\right) + c_2\beta_2\left(x^{globalbest} - x_i^{n-1}\right) \qquad (18)$$

Where $x_i^{localbest}$ best historical position, $x^{globalbest}$ best position of the entire population and c_1 and c_2 are impact factors for moving to local or global best answer.

4- Formulation of Proposed Firefly Algorithm

According to the concepts mentioned in Sections 3-1 and 3-2 ,the movement operator in Standard FA, Eq. (5), is change to Eq. (19).

$$\begin{aligned} x_i^n &= x_i^{n-1} + c_0\beta_0 e^{-\gamma r_{ij\in Kest}^2}\left(x_{j\in Kbest}^{n-1} - x_i^{n-1}\right) \\ &+ c_1\beta_1\left(x_i^{localbest} - x_i^{n-1}\right) + c_2\beta_2\left(x^{globalbest} - x_i^{n-1}\right) + ae_i^{n-1} \end{aligned} \qquad (19)$$

Where $x_i^{localbest}$ best historical position, $x^{globalbest}$ best position of the entire population , c_0 impact factors for moving to K-best of more attractive fireflies and c_1 and c_2 are impact factors for moving to local or global best answer.

Therefore, Eq. (19) provides the new definition of movement in Improved Firefly Algorithm. The flowchart of the Proposed Firefly Algorithm is shown in Fig. 2.

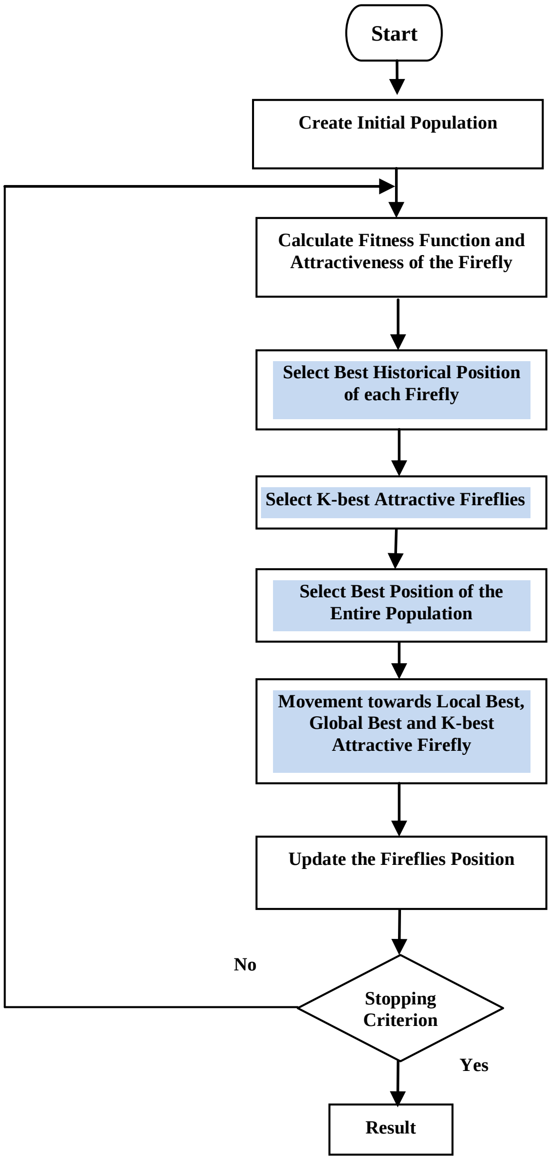

Start
Create Initial Population
Calculate Fitness Function and Attractiveness of the Firefly
Select Best Historical Position of each Firefly
Select K-best Attractive Fireflies
Select Best Position of the Entire Population
Movement towards Local Best, Global Best and K-best Attractive Firefly
Update the Fireflies Position
No
Stopping Criterion
Yes
Result

Figure 2. Flowchart of the Proposed Firefly Algorithm

5- Validation and Computational Experiment

To show the effectiveness and the power of Improved firefly algorithm, proposed in this paper, it is evaluated on minimizing 7 well-known benchmark functions [23].

These benchmark functions are presented in table 1. The number of variables in these objective functions is n=30.

Table 1. The 7 Benchmark Functions used in experimental study, n=30, [18]

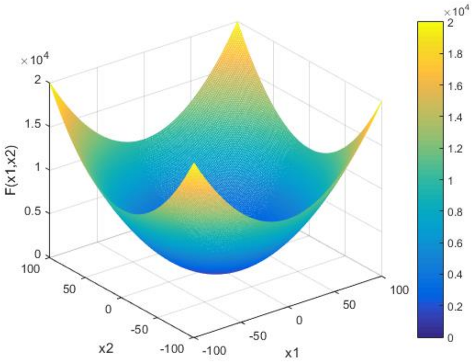
Name Function	Function	Range
Sphere Model	$F_1(x)=\sum_{i=1}^{n} x_i^2$	[-100,100]
Generalized Rastrigin's Function	$F_2(x)=\sum_{i=1}^{n}\left(x_i^2-10\cos\cos\left(2\pi x_i\right)+10\right)$	[-5.12,5.12]
Generalized Griewank Function	$F_3\left(x\right)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600,600]
Generalized Penalized Functions	$F_4(x)=\frac{\pi}{n}\times\{10\sin^2(\pi y_1)+$ $\sum_{i=1}^{n-1}(y_i-1)^2(1+10\sin^2(\pi y_{i+1})+(y_n-1)\ \}+$ $\sum_{i=1}^{n}u(x_i,10,100,4)$ $y_i=1+\frac{1}{4}(x_i+1)$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i \text{ f } a\\ 0 & -a\le x_i\le a\\ k(-x_i-a)^m & x_i \text{ p } -a\end{cases}$	[-50,50]
	$F_5(x)=0.1\{\ \sin^2(3\pi x_1)+$ $\sum_{i=1}^{n-1}(x_i-1)^2(1+\sin^2(3\pi x_{i+1})+$ $(x_n-1)(1+\sin^2(2\pi x_n)\ \}+\sum_{i=1}^{n}u(x_i,5,100,4)$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i \text{ f } a\\ 0 & -a\le x_i\le a\\ k(-x_i-a)^m & x_i \text{ p } -a\end{cases}$	[-50,50]
Ackley's Function	$f_6(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	[-10,10]
Schwefel's Problem	$f_7(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-$ $\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	[-32,32]

For the computational test purpose, the simulation executed on PC with Intel Core i5-2410M CPU @ 2.30 GHz processor and 6 GB RAM.

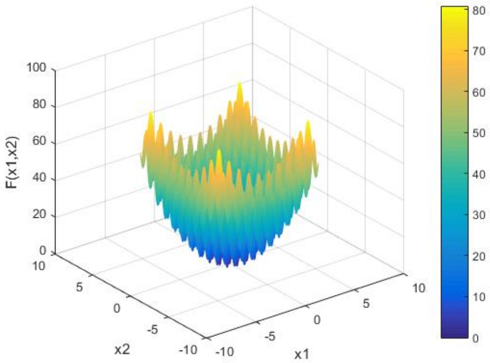
In the functions given on the first column of the table 1, the goal of the algorithm is to minimize the value of the function. On the other hand, this minimization must happen in the shortest possible time. The second column of the table shows the dimensions of the problem and the third column shows its limitations.

The constraints of the problem show the upper and the lower limit of the decision variables. Therefore, the algorithm must be find the best optimal answer with these limitations.

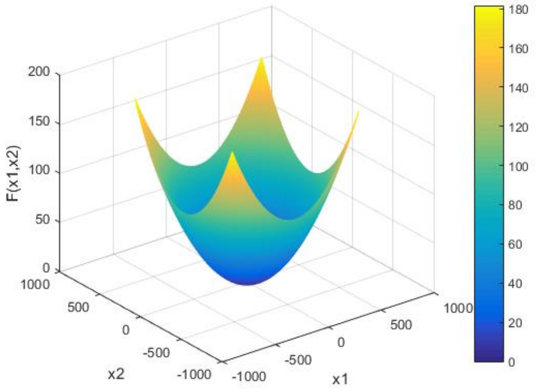
Graphs of three example functions (F1, F2 and F3) in two dimensions are also illustrated in Fig. 3.



A. Graph of F1



B. Graph of F2



C. Graph of F3

Figure 3. Graphs of three example functions (F1, F2 and F3) for n=2

In this section, five scenarios are defined to validate the performance of the Improved Firefly Algorithm obviously.

Scenario I. Standard Firefly Algorithm Simulation (SFA)

Standard FA is simulated in scenario I, initially. With this simulation, the initial state of the optimization is obtained.

In this scenario, Eq. (5) is used to obtain the new position of the firefly.

Scenario II. Applying K-best Attractive Fireflies in the Firefly Algorithm (K-FA)

In this scenario, for movement of firefly, each of them is attracted to K-best more attractive instead of all more attractive fireflies. Eq. (15) gives the new position of the firefly. Total running time of this method is expected to be reduced.

Scenario III. Applying Local Best Position of Each Firefly in the Firefly Algorithm (L-FA)

As mention in section 3-2, local best position of each firefly in movement FA, can be help to optimization process. In this scenario, movement FA is affected by two factors, their local best position and attractive firefly.

Scenario IV. Applying General Best position of Fireflies in the Firefly Algorithm (G-FA)

Similar to the scenario III, but with a difference, the global best position being used instead of local best position.

Scenario V. Applying K-best, local and General Best Position in the Standard Algorithm (IFA)

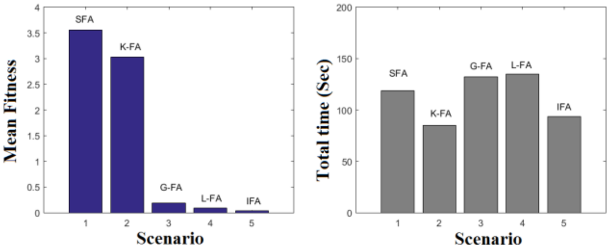
Finally, in scenario V, using results of scenarios II-IV, all parameters (K-best, local and global best position) are contributed to the optimization. This is proposed algorithm and it is expected to improve Firefly algorithm. In this scenario, Eq. (19) is used to obtain the new position of the firefly.

In the Following, scenario 1-5 are implemented on the test functions described in table I. In entire scenarios, population size is set to 40 and all of the fireflies are located in search space randomly. Maximum iteration of algorithm is 200. the result, that shown in tale 2, are achieved by the mean of 30 independent run for each scenario. *Kbest* in scenario II and V (K_FA and IFA) set to 50% of firefly population. Simulation results are shown in table 2 and figure 4. they are compared by Standard Firefly (scenario I, SFA).

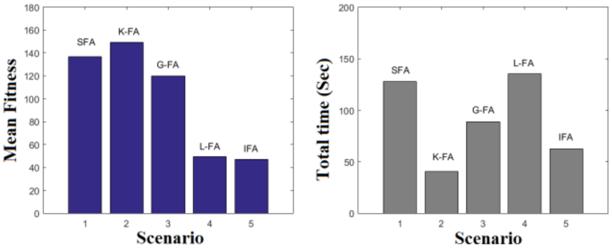
Table 2. The average of final best fitness and total time for 30 runs of minimizing benchmark functions 1, 2,3,4,5,6 and 7, number of iterations=200, n=30

Scenario		SFA	K_FA	G_FA	L-FA	IFA
Min F_1(x)	Mean Fitness	3.56	3.03	0.19	0.09	0.04
	Improvement		23.9%	94.6%	97.5%	98.9%
	Total Time	118.78	84.82	132.17	134.89	93.45
	Improvement		28.6%	-11.3%	-13.6%	21.3%
Min F_2(x)	Mean Fitness	136.87	149.20	119.8	49.32	47.06
	Improvement		-9%	12.5%	64%	65.6%
	Total Time	127.65	40.24	88.96	135.49	62.27
	Improvement		68.5%	30.3%	-6.1%	51.2%
Min F_3(x)	Mean Fitness	0.52	0.52	0.1	0.08	0.05
	Improvement		0%	80.8%	84.6%	90.4%
	Total Time	104.99	60.34	100.60	129.38	84.34
	Improvement		60.3%	4.2%	-23.2%	20%
Min F_4(x)	Mean Fitness	14.02	9.85	7.38	0.73	0.22
	Improvement		29.7%	47.4%	94.8%	98.4%
	Total Time	214.54	58.15	167.65	188.37	75.00
	Improvement		72.9%	21.9%	12.2%	65%
Min F_5(x)	Mean Fitness	44.56	41.34	3.14	-0.23	-0.5

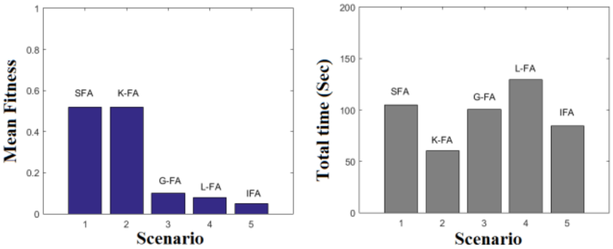
	Improvement		7.2%	93%	100%	101%
	Total Time	144.81	96.65	156.73	172.21	99.44
	Improvement		33.3%	-8.2%	-18.9%	31.3%
Min $F_6(x)$	Mean Fitness	35.47	28.26	27.52	4.61	4.81
	Improvement		20.3%	22.4%	87%	86.4%
	Total Time	146.86	95.43	153.33	150.77	107.23
	Improvement		35%	-4.4%	-2.7%	27%
Min $F_7(x)$	Mean Fitness	6.96	6.12	1.96	0.80	0.92
	Improvement		12.1%	71.8%	92%	90.8%
	Total Time	151.86	103.69	100.24	157.55	106.70
	Improvement		31.7%	34%	-3.7%	29.7%



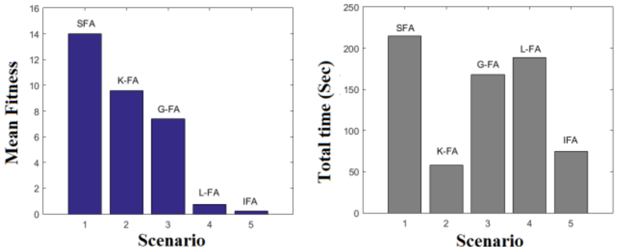
A. Mean Fitness Function and Total Running Time of F1



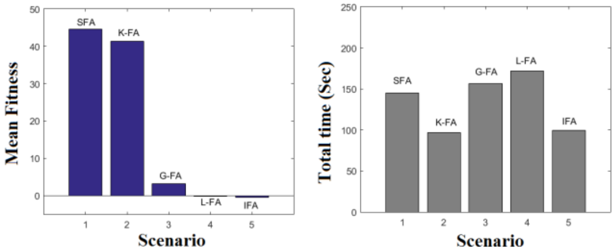
B. Mean Fitness Function and Total Running Time of F2



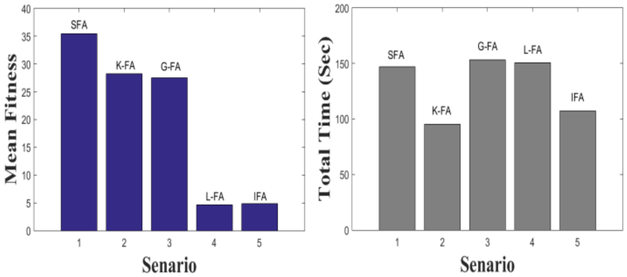
C. Mean Fitness Function and Total Running Time of F3



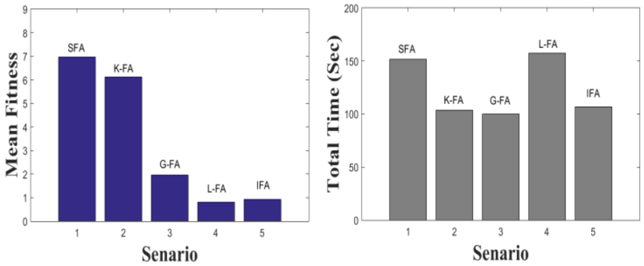
D. Mean Fitness Function and Total Running Time of F4



E. Mean Fitness Function and Total Running Time of F5



F. Mean Fitness Function and Total Running Time of F6



G. Mean Fitness Function and Total Running Time of F7

Figure 4, Mean Fitness Function and Total Running Time of Benchmark Functions

As is clear from the simulation results, performance of all the algorithms introduced in Scenario 2-5, (K_FA, G_FA, L_FA and IFA), are better than the Standard Firefly Algorithm (SFA). In K_FA, The total time of the 30 independent run is significantly reduced compared to the SFA. Of course, response accuracy is improved in most functions, too. On the other hand, By running the simulation, the response accuracy in G_FA and L_FA have increased and these scenarios have better performance than the SFA. As a result, in scenario 5, which is obtained from the integration of scenarios 2–4, both of the total time and the response accuracy has improved simultaneously compared to SFA. This improvement (IFA) is better than the other scenarios (K_FA, G_FA and L_FA).

6- Conclusion

This paper proposed models for improvement of Firefly algorithm. These models are developed using the Gravitational Search Algorithm (GSA) and Particle Swarm Optimization (PSO). K-best attractive fireflies, local best position of each firefly and general best position of population, are innovation that apply on Standard Firefly Algorithm. These parameters affect to the movement of fireflies and they cause the search process to be smart and fast. After analyzing and simulating the models, it is shown that they perform better than the standard version of the Firefly Algorithm. It is also found that among these models, performance of IFA is better than the other models and Standard Firefly Algorithm in accuracy and response speed.

Reference

[1]. N. Bacanin, M. Tuba, "Firefly Algorithm for Cardinality Constrained Mean-Variance Portfolio Optimization Problem with Entropy Diversity Constraint", Scientific World Journal, Vol. 2014, P.P. 1-16, May 2014.

[2]. W.W. Hwu, "GPU Computing Gems Jade Edition", Morgan Kaufmann press, 2011.

[3]. X.S. Yang, "Nature-Inspired Metaheuristic Algorithms", Luniver Press. 2008.

[4]. N. F. Johari, A. M. Zain, N.H. Mustaffa, A. Udin, "Firefly Algorithm for Optimization Problem", Applied Mechanics and Materials, Vol. 421, P.P. 512-517, 2013.

[5]. A.H. Damia, M. Esnaashari, "Automated Test Data Generation Using a Combination of Firefly Algorithm and Asexual Reproduction Optimization Algorithm", International Journal of Web Research, Vol. 3, No. 1, P.P. 19-28, Spring-Summer, 2020.

[6]. I. Brajević, P.S. Stanimirović, S. Li, X. Cao, "A Hybrid Firefly and Multi-Strategy Artificial Bee Colony Algorithm", International Journal of Computational Intelligence Systems, Vol. 13, P.P. 810-821, 2020.

[7]. F. Wahid1, M. Sultan zia, R.N. Bin Rais, M. AAmir, U. Muneer butt, M. Ali, A. Ahmed, I. Ali khan, O. KHalid, "An Enhanced Firefly Algorithm Using Pattern Search for Solving Optimization Problems", IEEE Access, Vol. 8, P.P. 148264-148288, 2020.

[8]. W. Alomoush, K. Omar, A. Alrosan, Y. M. Alomari, D. Albashish, A. Almomani, "Firefly photinus search algorithm", Journal of King Saud University – Computer and Information Sciences, Vol. 32, P.P. 599–607, 2020.

[9]. SH. Mashhadi Farahani, A.A. Abshouri, B. Nasiri, M.R. Meybodi, "Some Hybrid Models to Improve Firefly Algorithm Performance", International Journal of Artificial Intelligence, Vol. 8, P.P. 1-20, 2012.

[10]. G.G. Wang, L. Guo, H. Duan, H. Wang, "A New Improved Firefly Algorithm for Global Numerical Optimization", Journal of Computational and Theoretical Nanoscience, Vol. 11, P.P. 477–485, 2014.

[11]. F. Wahid, R. Ghazali, L.H. Ismail, "Improved Firefly Algorithm Based on Genetic Algorithm Operators for Energy Efficiency in Smart Buildings", Arabian Journal for Science and Engineering, Vol. 44, P.P. 4027-4047, 2019.

[12]. J. Nayak, B. Naik, P. Dinesh, K. Vakula, P.B. Dash, "Firefly Algorithm in Biomedical and Health Care: Advances, Issues and Challenges", SN Computer Science. Vol. 1, P.P. 1-36, 2020.

[13]. H. Zhang, J Yang, J. Zhang, P. Song, X. Xu, "A Firefly Algorithm Optimization-Based Equivalent Consumption Minimization Strategy for Fuel Cell Hybrid Light Rail Vehicle", Energies, Vol. 12, P.P. 1-18, 2019.

[14]. M. Zile, "Routine Test Analysis in Power Transformers by Using Firefly Algorithm and Computer Program", IEEE Access, Vol. 8, P.P. 132033-132040, 2019.

[15]. S.P. Mishra, P.K. Dash, "Short-term prediction of wind power using a hybrid pseudo-inverse Legendre neural network and adaptive firefly algorithm", Neural Computing and Applications, Vol. 31, P.P. 2243–2268, 2019.

[16]. X.S. Yang, "Firefly Algorithm for Multimodal Optimization", SAGA 2009, Lecture Notes in Computer Science, Vol. 5792, P.P. 169-178, 2009.

[17]. Rashedi, "Gravitational search algorithm," M.Sc. thesis, Dept. Elect. Eng., Shahid Bahonar University of Kerman, Kerman, Iran, 2007.

[18]. E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, "GSA: A gravitational search algorithm," Information Science, Vol. 179, P.P. 2232-2248, June 2009.

[19]. P. Tharawetcharak, T. Karot, C. Pornsing, "An Improved Gravitational Coefficient Function for Enhancing Gravitational Search Algorithm's Performance", International Journal of Machine Learning and Computing, Vol. 9, P.P. 261-266, June 2019.

[20]. R.C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, P.P. 39–43, 1995.

[21]. J. Kennedy, R.C. Eberhart, "Particle swarm optimization", in Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, P.P. 1942–1948, 1995.

[22]. Y. Jiang, T. Hu, C.C. Huang, X. Wu, "An improved particle swarm optimization algorithm", Applied Mathematics and Computation, Vol. 193, P.P. 231–239, 2007.

[23]. X. Yao, Y. Liu, G. Lin, "Evolutionary Programming Made Faster", IEEE Transactions on Evolutionary Computation, Vol. 3, No. 2, P.P. 82-102, 1999.

Mahdi Tourani is an Assistant Professor with the Department of Electrical Engineering, Technical faculty of Ferdows, University of Birjand, Birjand, Iran. He received his B.Sc. in Electrical Engineering from Mashhad Ferdowsi University, Iran, in 2008. He also received his M.Sc. and Ph.D. degrees in Electrical Engineering from University of Birjand, Birjand, in 2011 and 2016, respectively. His areas of interest include Computational Intelligence, Smart Grids, Micro Grids as well as Electrical Vehicles.

The Development of a Hybrid Error Feedback Model for Sales Forecasting

Mehdi Farrokhbakht Foumani*
Department of Computer Engineering, Islamic Azad University, Fouman and Shaft Branch, Fouman, Iran
M.Farrokhbakht@fshiau.ac.ir
Sajad Moazami Goudarzi
Department of Computer Engineering, Islamic Azad University, Tehran North Branch, Tehran, Iran
SajadMoazami@yahoo.com

Received: 08/Oct/2020 Revised: 17/Apr/2021 Accepted: 11/May/2021

Abstract

Sales forecasting is one of the significant issues in the industrial and service sector which can lead to facilitated management decisions and reduce the lost values in case of being dealt with properly. Also sales forecasting is one of the complicated problems in analyzing time series and data mining due to the number of intervening parameters. Various models were presented on this issue and each one found acceptable results. However, developing the methods in this study is still considered by researchers. In this regard, the present study provided a hybrid model with error feedback for sales forecasting. In this study, forecasting was conducted using a supervised learning method. Then, the remaining values (model error) were specified and the error values were forecasted using another learning method. Finally, two trained models were combined together and consecutively used for sales forecasting. In other words, first the forecasting was conducted and then the error rate was determined by the second model. The total forecasting and model error indicated the final forecasting. The computational results obtained from numerical experiments indicated the superiority of the proposed hybrid method performance over the common models in the available literature and reduced the indicators related to forecasting error.

Keywords: Data mining; Machine learning theory; Supervised learning; Sales forecasting.

1- Introduction

Data processing and analysis has been recognized as an efficient and lucrative process for organizations over the recent years. Thus the data mining methods that discovers and extracts useful patterns from such large data sets to find hidden and worthy patterns for the decision-making as well as machine learning theory methods have turned out to be very useful[1]. Sales forecasting is known as one of the most important applications of these methods. Sales forecasting is the process of determining the future demand of customers which can be conducted in the short term or long term. In sales forecasting, it is determined how much sales the sales team or participation at a certain time period (weekly, monthly, quarterly, or annually) achieves. Managers use sales forecasting in all of their agencies to estimate the amount of conducted transactions by their whole team. In addition, they use the forecasting conducted by the sales team to forecast the total amount of sales in that part of the organization[2].

A variety of methods have already been proposed for sales forecasting. These methods have generally been developed using two general approaches[3]:

1- Approaches related to time series forecasting. In this approach, the existing models focus on the demands that contain seasonality patterns. In the absence of a regular seasonality pattern, time series forecasting methods will not be very successful.

2- Calendar information that influences demands and consequently the forecasting procedures using the approaches derived from machine learning theory. These approaches have been successful in some businesses such as retailers, airlines, car sales, etc., and are generally preferred over time series approaches.

Researchers believe that sales forecasting is indispensable from the following perspectives[4]:

A) Guiding the chain towards the point where the inventory level is optimized and the products needed to meet customer demands and needs are not in short supply.

B) Facilitation of capacity planning, warehouse placement, production planning, procurement and prediction of raw materials required in the production stage.

C) Facilitation of matters related to sales and operations planning

According to the literature, companies with accurate sales forecasts are characterized by smaller warehouse space (by 15 %), higher OTIF index (by 17%) and lower cash-to-cash cycle time (by 35%)compared to other companies[5].

Taking into account the benefits of implementing sales forecasting programs, the need to design algorithms for this purpose is highlighted. Thus, the current research proposal seeks to develop an algorithm for short-term sales forecasting.

Accordingly, achieving a sales forecasting method with acceptable accuracy requires to consider all the effective factors and is regarded as a complicated process. In this regard, this study uses the two-step error feedback algorithm to present a new method for sales forecasting where the forecasted error values were corrected by a forecasting algorithm and thus the model error was reduced.

2- Review of literature

Sales forecasting can be largely significant in the success and performance of organizations. Forecasting with accuracy may lead to the lost demand or increase the product durability. Sales forecasting is one of the complicated problems in analyzing time series and data mining due to the number of intervening parameters. The factors such as population, purchase behavior, customers' tastes, and competitors' performance affect the product sales. In addition, the factors such as holidays, climatic conditions, daily events in society, economic status, and some unpredictable factors affect the customer purchase [4]. Another challenge is originated from the fact that fashion products are mostly saturated at a relatively short time and replaced by another fashion product leading to the lack of historical data for sales [6].

Sohrabi et al.[7] in their study, attempted to develop accurate classification algorithms (e.g., decision tree, art neural networks, support machines, and logistic regression).

For sales forecasting, the statistical methods such as exponential smoothing, ARIMA, Box and Jenkins, Regression, and Holt-Winters models were used. However, some advanced models such as neural network, support vector regression, and other hybrid models of data mining were used in recent years [8].

As the introduction and brief review of the related studies mentioned, Sun et al.[9] used extreme learning machine (ELM) for clothing sales forecasting. Their study followed up the study by Zhu et al.[10] using extreme learning machine as a fast method and introducing it as appropriate for bulk data.

Wong and Guo [11] combined extreme learning machine to harmony search algorithm and indicated that such a combination improves the accuracy of extreme machine in sales forecasting in the retail industry. In the proposed method, the harmony search algorithm has the task of training the extreme machine. Parameter turning is conducted by using a heuristic method.

Yu et al. [12] used the support vector machine regression to forecast the number of newspaper daily sales for a newspaper and magazine publication company. In this study, the support vector machine linear model and its nonlinear model were used by combining the rapid basis kernel function where the rapid basis model has a higher accuracy. Di pillo et al.[13] in another study explained the use of support vector regression for sales forecasting and used this method in another case study. Table 1 indicates some areas where machine learning methods and data mining are used for sales forecasting.

Table 1: Different areas in sales forecasting

Researcher(s)	*year*	*Area of study*
Aramaki et al. [14]	2011	Epidemiology
Asur&Haberman[15]	2010	Mailbox sales profits
Bollen et al. [16]	2011	Supermarket sales
Choi & Varian [17]	2012	Ticket sales on a tour
Dhar& Chang [18]	2009	Online music sales

The models in sales forecasting are divided into three main groups based on the focus and objective of the developed model.

1. The models related to sales variable: This type of models makes predicting models by considering all sales factors such as calendar data such as calendar information, special discounts, the plans related to customers' club, etc. to the high number of variables. In this regard, the methods of feature selection are widely used and improve the model speed.

2. The models related to the sales dependency of the product group: In such models, the method used for forecasting each product considers the sales of other products and discovers the dependency between the sales of different products. In the forecasting phase, the future sales value for a specific product group is determined according to the sales of other products. For example, the sales of macaroni and tomato paste have an effect on each other due to high dependency. For this reason, the forecasted sale value is considered for the other product group while forecasting the sale of one of them and finally will be effective on the final value of the forecasting.

3. Stock-keeping unit models: The models related to this area forecast the sales value of each stock-keeping unit (a specific barcode) instead of forecasting the sales of the product group. The sales fluctuations of a specific barcode are more than a product group and thus the sales forecasting for that is more difficult than the forecasting of the product group. However, the efficiency of such models is more useful to the organization in terms of implementation.

Here is the study of the sales of forecasting models.

2-1- The Models Related to Sales Variables

The models related to the variables affecting the sales of a product examine the effective factors in the product sales and perform sales forecasting by participating them in modeling[19]. The statistical models such as regression method are very useful in this regard. Since the methods related to this area usually involve many variables and lead to the reduction of algorithm processing speed, the feature selection techniques are required to remove the variables with less degree of significance or select the best subset of the current variables. Based on the study of John et al. [20], selecting a subset of the best variables is placed on in the class of Np-hard problems. For this reason, feature selection is conducted heuristically through two approaches of forward selection or backward elimination. These two approaches were developed in form of the algorithms such as genetic algorithm and simulated annealing[21]. Using such algorithms can converge to optimal subsets at a short time.

2-2- The Models Related to The Sales Dependency of The Product Group

Since buying some products may depend on some other products, the models related to the dependency of groups can be used as the efficient methods. The issue of demand dependency of different products was created in economics where the supplement and alternative products affect each other. Thus, the information related to changing one of them can be used in forecasting the demand of other products. In a product group, different products with different weights and packaging play the role of alternative products[22], [23]. The findings of researchers indicated that a large part of purchase behavior changes at the discount time is due to the presence of dependency in buying different products [24].

In most cases, buying a product available in the store or a product with discount, leads to the purchase of other products.

For example, the presence of discount on a specific kind of cake will lead to the increase of purchase on the complete package of different cakes. This type of dependency is known as intra group. As another example, the presence of discount on a type of macaroni may increase the sales of tomato paste. This is an inter group dependency. The effect of intra group and inter group dependencies leads to the increase of selling the products without discount.

Considering the inter group and intra group dependencies will lead to the increase of complexity in the final model. For example, in the VAR method, the number of variables increases as a quadratic function leading to the final complexity of the model. In such cases, even simple linear models cannot be used due to the multiple variables. Figure 1 indicates the multiplicity of dependency among different product groups [25].

2-3- The Forecasting Models of The Stock-keeping Unit

The models related to the stock-keeping unit are usually referred to the single variable models which forecast the future sales in the short term using time series analysis [26].

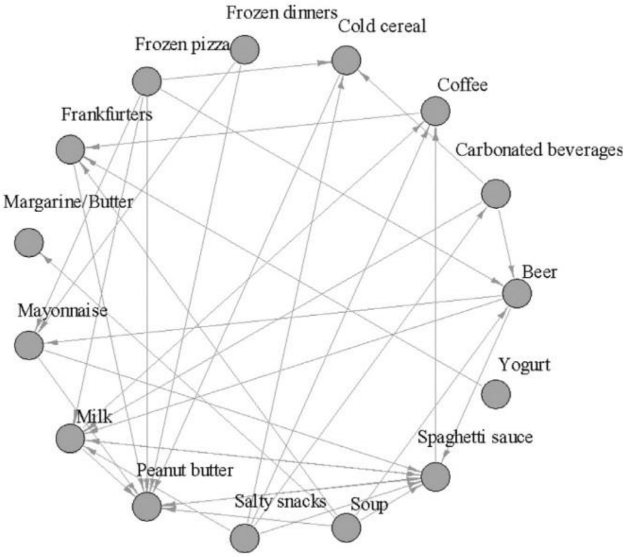


Fig. 1 The inter group dependency in making the forecasting models

Such methods do not consider the external variables such as the changes of price or discount plans. Goorali et al. [27] stated that the methods related to time series for the periods without discount work better than the other methods and show less error. However, for the periods where the policy of discount is considered, the methods such as regression entering external factors in the model will have superiority. Baeke et al. [28] presented a model where judgement and opinion of managers are combined to statistical models and finally the future demand value is forecasted.

Two approaches were presented for considering the managers in the used statistical model. The restrictive model where the forecasting of managers is presented as a range and used as the upper and lower limits of future forecasting in the model as well as the integrative model where the forecasting of managers is entered into the model as a new variable.

In this regard, Li and Lim [29] introduced a decomposition- aggregation algorithm and used it as the sales forecasting of clothing in Singapore. Their proposed method used the analysis of previous process of each item and the total behavior of sales in a store to conduct the forecasting. The comparison to previous methods confirmed the superiority of decomposition- aggregation algorithm.

The other related methods can be searched in the studies by Kourentzes and Petropoulos [30] and Babai et al. [31].

3- The Proposed Model

Using the Sine models in modeling the time series was introduced by Simons [32] and its general relation is as Equation (1).

$$\{Y_t\} = a_0 + \sum_{j=1}^{n} \left(a_j \cos\left(\lambda_j t\right) + b_j \sin\left(\lambda_j t\right) \right) \tag{1}$$

Where a_j and λ_j are the parameters related to the model and should be adjusted accurately for the access to the acceptable model. Using this model can be an alternative to the time series classic decomposition model.

Ord and Fildes [26] use the Sine model and multiple regression model to develop an algorithm for forecasting the time series being associated with the improved accuracy of process forecasting and seasonal model of time series. However, the main problem of the proposed method is the lack of paying attention to the external variables which may affect time series at specific times. For example, the presence of lunar occasions or official holidays in case of product sales are among these variables. Accordingly, the present study combined the multiple regression method (considering the external variables based on Table 2) to a Sine feedback model to forecast the sales. In other words, the proposed sales forecasting model is a hybrid algorithm correcting the conducted forecasting based on the feedback of error values. In this regard, after the initial training, the error values are determined and entered into another algorithm as a time series. Then, the second algorithm is trained based on the error values. For forecasting the future sales, first the time series related to the sales on previous days are entered into the model and an initial forecasting is obtained. Then, the initial forecasting is entered into the second algorithm and the final correction is made based on the correction factor obtained in the training group. Figure 2 shows a general view of the proposed hybrid algorithm.

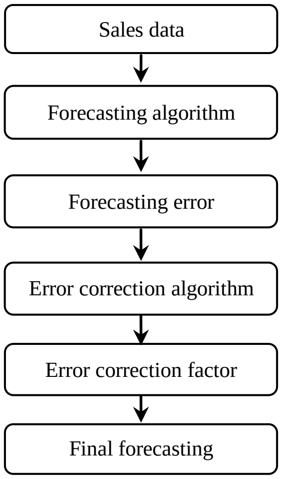


Fig. 2 The hybrid error feedback algorithm

3-1- Problem Data

The data used for constructing the proposed hybrid model and measuring its performance were related to the sale of 1850 items in one of the chain stores of Iran collected during 400 days. Since the information about the inventory is not available and the zero inventory may result in the non-sale of the product in relation to zero sale in the store, the proposed model can be only implemented for the products with at least 300 days of positive sale (At least 75% of the total days). The variables used for determining the amount of sale and extracting the related pattern were only of calendar variable type. In this regard, Table 2 indicates the variables used for constructing the sale forecasting model.

Table 2:The variables used in problem modeling.

Variable name	***Variable type***	***Description***
Season	Categorical	It represents the season of the year.
Month	Categorical	It represents the month of the year.
Week	Categorical	It indicates the number of the week including the numbers from 1 to 4.
Day	Categorical	The numbers 1 to 31 indicate the date of the day. Since the sale increase at the end of the month due to discounts, this variable can control some parts of the sale fluctuations.
Off day	Binary	Zero-one variable indicates if a particular day is off or not.

Variables in the present study can be divided into two groups: independent variables and dependent variables. According to the literature on sales forecasting, the independent variables that can affect sales are as follows:

1- Year: This variable can be used to discover the pattern in the sales trends.

2- Season: Some seasonal patterns can be extracted based on the level of sales in different seasons of the year.

3- Month: Since product sales may vary from one month to another, this variable will also be taken into account in the model.

4- Week days: Product Sales are more dramatic on busy shopping days (usually weekends) than week days. Thus, the number of week days will also be taken into account as a variable.

5- Holiday: Variable 0 and 1 indicate whether or not a particular day is holiday.

The dependent variable in this study is the quantity of sales (number of products sold)on a particular day.

3-2- Outlier Detection

Among the above-mentioned data, sometimes the sale values have an emotional growth which should be detected and eliminated at the corresponding time series. For example, some products such as milk face an emotional growth during the days when the air pollution increases due to inversion. Since the weather condition is not available because of the current limitations and cannot be entered into the model as a variable, the presence of the data related to emotional growth will lead to an increase in the model error. As another example, the products such as salt or cheap biscuits are sometimes offered to customers as free which increases the sale and the effect of free supply cannot be included in the forecasting model. For this reason, the outlier should be detected and eliminated at the time series. In this regard, the methods of outlier detection has been checked out and the most appropriate method for use in the proposed model is described.

In general, outliers to data that are much smaller or larger than other members of a given dataset. Outliers can affect the results and in fact impair their accuracy. Therefore, such data need to be analyzed and deleted from the dataset(if any)[33].

Depending on the number of variables, outlier detection techniques are categorized in two groups: univariate and multivariate detection techniques. In former case, only one variable is taken into account and a value is assigned to them based on their placement in the outlier region (univariate techniques) in the latter case, however, the variables in the data set are taken into account and addressed simultaneously. (multivariate techniques).

A method called Hempel technique is used for filtering the wave in the topics related to signal analysis. Hempel technique is appropriate for the cases where the value of a variable is collected over time and time intervals can be continuous (momentary) or discrete. Since the time series related to the sale of products is a wave with discrete time intervals (a quantity for sale every day). Thus, the present study will use Hempel technique.

For more explanation, consider a time series as Equation (2).

$$\omega_k^K = \{ x_{k-K}, ..., x_k, ..., x_{k+K} \} \tag{2}$$

The mean value of this time series is indicated as m_k and defined as follows:

$$m_k = median\ \{ x_{k-K}, ..., x_k, ..., x_{k+K} \} \tag{3}$$

Accordingly, the Hempel detector variable $y_k \in \{ x_k, m_k \}$ will be a binary variable for which m_k value means the k -th in the time series introduced time series in Equation (3) is an outlier. Thus, its median value will be replaced, otherwise the above-mentioned data cannot be considered as outlier. The detector variable $y_k \in \{ x_k, m_k \}$ is defined as Equation (4).

$$y_k = \begin{cases} x_k & |x_k - m_k| \le t.s_k \\ m_k & |x_k - m_k| > t.s_k \end{cases} \tag{4}$$

Where the estimator is the Hampel median coefficient following Equation 5.

$$s_k = 1.4826 \times \underset{j\in[-k,k]}{median} \left\{ \left| x_{k-j} - m_k \right| \right\} \tag{5}$$

In addition, parameter t is known as the decision boundary and is typically considered in such a way that the following equation is true:

$$t \ge \frac{\max_k |x_k - m_k|}{\min_k s_k} \tag{6}$$

3-3- Hybrid error Feedback Algorithm

As mentioned in the current section, the proposed error feedback model includes two algorithms one of which is used for the initial sales forecasting and the other one for correcting the error values. In this procedure, first a clustering algorithm is used on the time series related to the sales of used items and a forecasting is obtained. An initial forecasting is obtained using the created model and the forecasted error values are stored. For more explanation, assume that there is a time series as $\Xi = \{ x_1, x_2, ..., x_n \}$ for one of the items in the store where x_i represents the sales amount on the i-th day. By considering the above-mentioned time series, the first clustering algorithm considers the calendar data introduced in Table 3 as the input and the sales amount as the output and then is trained. After training, an initial sales forecasting is obtained using the same inputs. The initial forecasting value for sales is named $\hat{y}_i$. Since the model may have errors, the error value is calculated using the initial forecasting through Equation (7).

$$e_i = \hat{y}_i - x_i \tag{7}$$

Where e_i represents the error related to the model.

In the second group, the inputs related to the calendar data based on table 2. (similar to the first algorithm) are entered into the second algorithm (error corrective algorithm) and then the values related to error are considered as the output.

In fact, the second algorithm attempts to determine the error related to the daily sales. At the final step, the calendar data related to the future days are entered into the first algorithm. Then, an initial sales forecasting value $\hat{y}_i$ with the calendar data are entered into the second algorithm and the forecasting error value is specified $\hat{e}_i$. Finally, the sales forecasting value is calculated using Equation (8).

$$y_i = \hat{y}_i + \hat{e}_i \tag{8}$$

The flowchart related to the hybrid error feedback algorithm is shown as Figure 3.

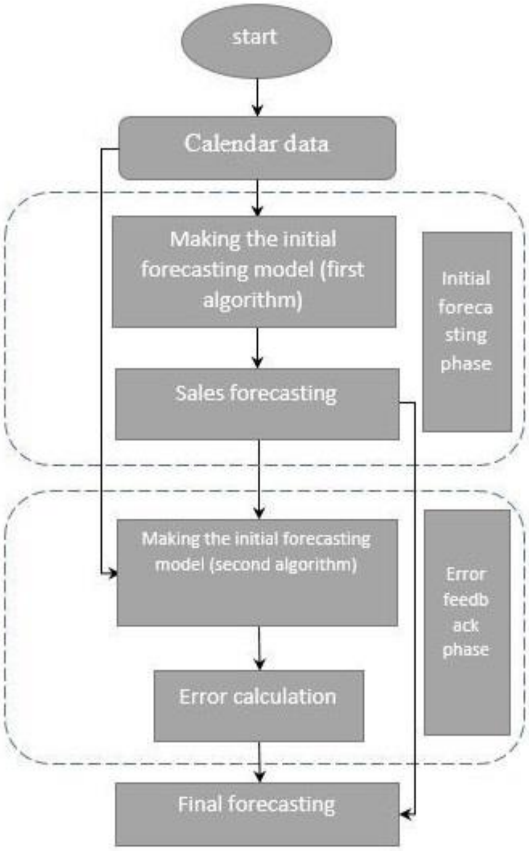

start
Calendar data
Making the initial forecasting model (first algorithm)
Initial foreca sting phase
Sales forecasting
Making the initial forecasting model (second algorithm)
Error feedb ack phase
Error calculation
Final forecasting

Fig. 3 The flowchart related to the hybrid error feedback algorithm

In the error feedback phase, the model error is estimated using a general relation according to Equation (9) having fewer parameters than the Simon's model [32].

$$\hat{e}_i = a_1 \sin\left(a_2 i + a_3\right) + a_4$$

(9)

In other words, the values related to the error are calculated after making the initial model and then considered as a dependent variable for Equation 3.10. Then, the best values for $\{a_1, a_2, a_3, a_4\}$ are calculated using model (10) called least square method.

$$\hat{e}_i = \sum_{i=1}^{n} \left(a_1 \sin\left(a_2 i + a_3\right) + a_4 - e_i\right)^2 \tag{10}$$

Where n represents the number of train data and i indicates the number of data.

By specifying the values $\{a_1, a_2, a_3, a_4\}$, Equation (9) is used and error estimate $\hat{e}_i$ is calculated. Finally, the final forecast is obtained by Equation $\hat{y}_f = \hat{y}_i + \hat{e}_i$.

4- Computational Results

All computations related to the proposed model were conducted in MATLAB R2017b software using a system with Core i3 processor and internal memory of 4Gb. Since the direct forecasting for each product leads to the creation of a lot of chaos in the models and cannot have enough accuracy, the proposed model is matched on the sales data of the product group. In this regard, the sales of each product in one group is added to each other and finally the sales of the product group are examined as the studied variable (e.g. the low fact milk group includes different products such as 500 CC and 1000 CC from different companies). Accordingly, 371 product groups are extracted from 1850 products in the dataset. In order to measure the performance of the proposed model and compare it to the previous models in the literature, the following parameters are used:

Root mean square of errors: If the real demand value on day t is equal to y_t and its forecasting value is equal to $\hat{y}_t$, the value of this index for n periods will be calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{n}} \tag{11}$$

Mean Absolute Errors:

$$MAE = \frac{\sum_{i=1}^{n} \left|y_i - \hat{y}_i\right|}{n} \tag{12}$$

Mean Absolute scaled Errors:

$$MASE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|y_i - \hat{y}_i|}{\frac{1}{n-1}\sum_{k=2}^{n-1}|y_k - y_{k-1}|}\right) \quad (13)$$

The variables related to the dataset include the calendar data such as year, month, day, week, and holidays. Accordingly, the sales amount is considered as the dependent variable and the forecast model is matched to it. In addition, other data of product are considered for reaching the sales forecasting of some. In this regard, the sales of similar products are classified in a variable called product group. For example, the sales of cola, beer, etc. will be classified under the name “carbonated beverage”. The sales of mineral water and other non-carbonated beverages is classified under the name “non-carbonated beverages” and forecasting is conducted for the level-4 sales of the product group. Finally, the share of each product from the total sales of product group is determined and accordingly, the final sales will be specified in terms of products.

In order to achieve an accurate dataset, the preprocessing should be conducted and some outliers should be eliminated from the studied dataset. For this reason, the Hampelfilteris used where the time series related to the sales is studied and the related outliers are eliminated. In this regard, Table 3 shows the data in relation to the number of eliminated data in each product group.

In addition, Figure 4 shows the frequency of outliers. This figure indicates that the maximum repetition of outliers is between 60 to 80 outliers being repeated for 113 product groups.

Table 3: The summary of data related to outliers

Title	*value*	**Product group**
the maximum number of outliers	206	Hand soap powder
the minimum number of outliers	2	Shaving foam
the average number of outliers in each group	73.29	-
The total number of outliers	19571	-

The proposed hybrid model includes two phases. In the first phase, the initial forecasting is conducted by using a forecasting model and then the errors related to the forecasting are determined, in the second phase, the values related to forecasting error are determined using an error corrective algorithm. Finally, the final output is determined by adding the initial forecasting and predetermined error.

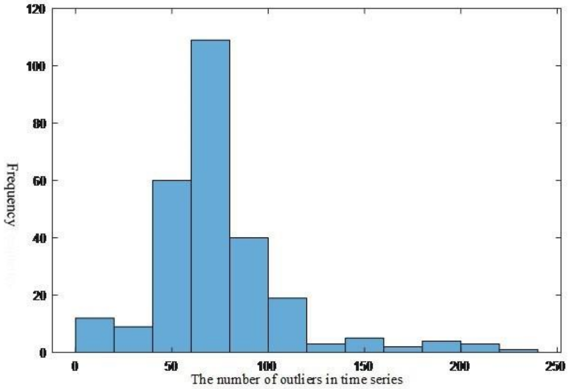


Fig. 4 The histogram related to the frequency of outliers

In this regard, for example the diagram related to the daily demand of minced meat is shown in Figure 5.

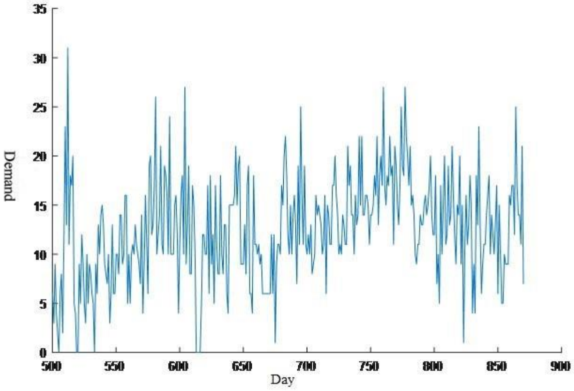


Fig. 5 The demand related to minced meat product group

First, the linear regression model is calculated using the calendar data in Table 2 based on the train data and then is used for test data. The output of the above-mentioned regression model will be the initial forecasting value $\hat{y}_i$ .Figure 6 shows the related algorithm.

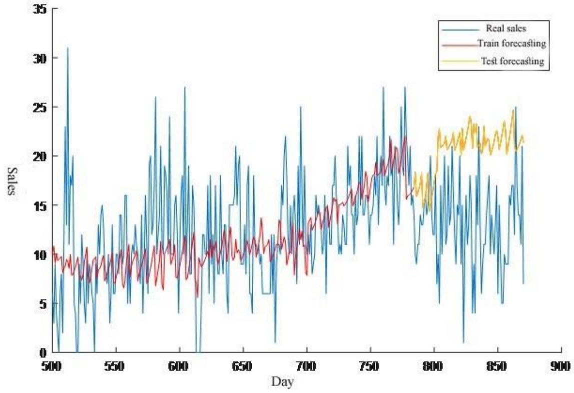


Fig. 6 The output related to the regression model as the initial forecast of minced meat product group

Accordingly, it is clear that the regression model is ascendingly trained due to the presence of an ascending trend on the sales data and caused some errors while forecasting the days related to the test set.

The related indices for this case are RMSE=9.19, MAE=7.96, and MASE=1.69.

Thus, due to the presence of error in discovering the models related to the time series of studied dataset, the error in the forecasting should be detected and simulated for correction by the Sine model introduced in the third chapter. After performing the simulation and training of the Sine model, Figure 7 is considerable.

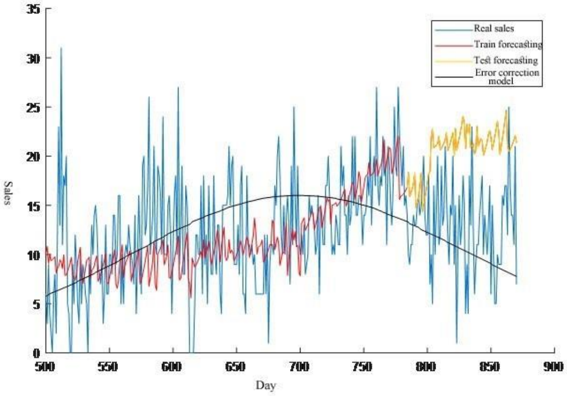


Fig. 7 The output related to the corrective model of error in minced meat product group

Obviously, the model shown in Figure 7 has the ability to discover the sales reduction than the trend related to the regression model.

Accordingly, the final value can be determined based on the initial forecast and output of the Sine model. Figure 8 shows the final forecasting.

It is graphically obvious that the final forecasting value is superior to the regression model.

In this relation, the correlation indicators are as RMSE=4.63, MAE=3.62, and MASE= 0.77.

Similar calculations are also conducted for other product groups and finally the obtained values are compared to the indicators related to the common regression model in the literature.

Accordingly, the results related to the calculations from the perspective of three indicators mentioned above are as Table 4.

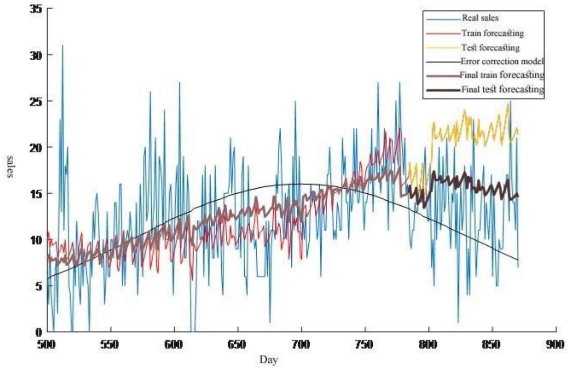


Fig. 8 The final forecasting related to the sales of minced meat product group

Based on Figure 9 and Table 4, it is clear that the proposed method in leads to the improvement of 13 product groups. The average value of improvement observed on all product groups is equal to 4.13 in terms of MAE.

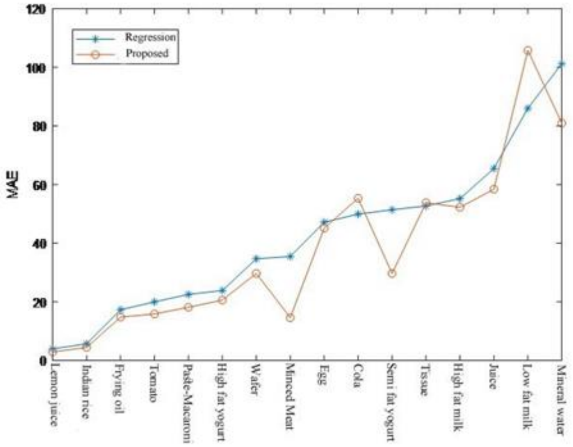


Fig. 9 Comparing the MAE performance indicator of the regression model and the proposed model

Table 4. The obtained results for the indicators of regression model performance, Ord and Fildes' model, and the proposed model

Product group	***Regression model***			***Ord and Fildes model (2003)***			***The proposed method***		
	MAE	RMSE	MASE	MAE	RMSE	MASE	MAE	RMSE	MASE
Lemon juice	3.99	4.85	1.06	3.03	4.29	0.83	2.94	3.89	0.78
Indian rice	5.71	8.10	1.17	5.35	7.17	0.96	4.48	7.01	0.92
Frying oil	17.32	21.46	1.01	16.36	20.78	1.02	14.81	19.53	0.86
Tomato paste	20.02	24.65	1.06	16.71	23.59	0.99	15.90	20.44	0.84
macaroni	22.56	26.68	0.96	14.92	22.29	0.73	18.16	22.53	0.77
High fat yogurt	23.89	27.75	1.34	23.78	27.85	1.22	20.63	24.36	1.16
Wafer	34.73	44.86	1.17	31.48	38.86	1.16	29.72	38.77	1.00
Minced meat	35.49	39.90	2.89	13.74	14.16	0.97	14.63	17.28	1.19
egg	47.18	52.83	2.32	50.34	54.80	2.56	45.18	50.66	2.22
soda	49.96	61.74	0.88	63.54	73.23	1.06	55.38	65.65	0.97
Half-fat milk	51.45	54.46	3.73	26.92	31.13	1.84	29.71	33.87	2.16
Tissue	52.73	69.82	0.71	56.28	78.77	0.80	53.85	73.27	0.72
High fat milk	55.24	76.59	0.91	61.63	81.56	0.99	52.29	75.25	0.86
Juice	65.51	88.79	0.83	65.53	84.39	0.80	58.44	84.25	0.74
Low fat milk	86.06	102.91	1.44	118.40	128.92	2.01	105.81	123.67	1.78
Mineral water	101.26	142.05	0.99	85.37	121.76	0.85	81.00	118.97	0.79
Average	42.07	52.97	1.40	40.84	50.85	1.17	37.68	48.72	1.11

5- Conclusion

The present study provided a new model by combining the regression forecasting algorithm and error feedback model

to forecast the sales of products in the stores where the initial forecasting was conducted using the regression model and then the remaining values (error) in the forecasting were corrected using an error feedback model to correct the initial forecasting values. In order to review the activities conducted in this study and based on the computational results, the proposed hybrid method had superiority over the common regression model in the available literature and reduced the indicators related to forecasting error. The improvement rate observed on 16 product groups was explained in detail and it was specified that the proposed method created 4.13 improvement units in MAE index. Other performance indicators of the model also indicated the improvement.

In order to improve the results and develop the introduced model, the following issues can be discussed in this chapter.

- Using the Gaussian kernel function due to high flexibility in modeling the complicated structures can lead to the improvement of the model performance. In this regard, the computation time may increase slightly. However, the increase of computational time can be ignored in case of improving the accuracy.
- Using other forecasting methods instead of regression method to improve the initial forecasting which can improve the final forecasting and increase the final accuracy.
- Using a nested approach which means combining the clustering and classification and using a forecasting model inside each cluster, by doing this, each classifier is used on the similar data and can increase the accuracy.

References

[1] B. Sohrabi, I. RaeesiVanani, N. Nikaein and S. Kakavand, "A predictive analytics of physiciansprescription and pharmacies sales correlation using data mining", International Journal of Pharmaceutical and Healthcare Marketing, vol.13, No.3, pp. 346-363, 2019.

[2] Y. Kaneko and K. Yada, "A Deep Learning Approach for the Prediction of Retail Store Sales", In Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference, 2016, December, pp. 531-537.

[3] M. Bohanec, M.K. Borštnar and M. Robnik-Šikonja, "Explaining machine learning models in sales predictions", Expert Systems with Applications, vol. 71, pp. 416-428, 2017.

[4] S. Thomassey, "Sales forecasts in clothing industry: The key success factor of the supply chain management", International Journal of Production Economics, Vol. 128, No. 2, pp. 470-483, 2010.

[5] A. Demiriz, "Demand Forecasting based on Pairwise Item Associations",Procedia Computer Science, vol. 36, pp. 261-268, 2014.

[6] S. Thomassey,"Sales forecasting in apparel and fashion industry: A review", Intelligent fashion forecasting systems:

Models and applications,Springer, Berlin, Heidelberg, pp. 9-27, 2014.

[7] B. Sohrabi, I. RaeesiVanani, A. Gooyavar and N. Naderi, " Predicting the Readmission of Heart Failure Patients through Data Analytics ", Journal of Information & Knowledge Management, Vol.18, No.1, pp.1950012-1, 1950012-20, 2019.

[8] N. Liu, S. Ren, T.M. Chio, C.L.Hui,andS.F.Ng,"Sales forecasting for fashion retailing service industry: a review", Mathematical Problems in Engineering, Vol.20, No. 2, pp. 22-29, 2013.

[9] Z.L. Sun, T.M. Choi, K.F. Au, and Y. Yu,"Sales forecasting using extreme learning machine with applications in fashion retailing", Decision Support Systems, Vol.46, No. 1, pp. 411-419, 2008.

[10] Q.Y.Zhu, A.K.Qin, P.N.Suganthan, and G.B.Huang, "Evolutionary extreme learning machine", Pattern recognition, Vol.38, No. 10, pp.1759-1763, 2005.

[11] W. K. Wong, and Z. X. Guo,"A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm", International Journal of Production Economics, Vol.128, No. 2, pp.614-624, 2010.

[12] X. Yu, Z. Qi, and Y. Zhao,"Support vector regression for newspaper/magazine sales forecasting", Procedia Computer Science,vol. 17, pp. 1055-1062, 2013.

[13] G. Di Pillo, V.Latorre, S. Lucidi, and E. Procacci,"An application of support vector machines to sales forecasting under promotions", 4OR, Vol.14, No. 3, pp. 309-325, 2016.

[14] E. Aramaki, S. Maskawa, and M. Morita,"Twitter catches the flu: detecting influenza epidemics using Twitter", In Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, N. Eight Street, Stroudsburg, PA, 18360 United States, 2011, pp. 1568-1576.

[15] S. Asur and B.A. Huberman,"Predicting the future with social media", In 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010, IEEE, Toronto, Canada,Vol.1, pp. 492-499.

[16] J. Bollen, H. Mao, and X. Zeng,"Twitter mood predicts the stock market", Journal of computational science, Vol. 2, No. 2, pp. 1-8, 2011.

[17] H. Choi, and H. Varian,"Predicting the present with Google Trends", Economic Record, Vol.88, No. S1, pp.2-9, 2012.

[18] V. Dhar, and E.A. Chang,"Does chatter matter? The impact of user-generated content on music sales", Journal of Interactive Marketing, Vol.23, No. 4, pp.300-307, 2009.

[19] D. L. Donoho,"High-dimensional data analysis: The curses and blessings of dimensionality", AMS math challenges lecture, pp. 1-32, 2000.

[20] G. H. John, R. Kohavi, and k. Pfleger,"Irrelevant features and the subset selection problem", In Machine Learning Proceedings, Proceedings of the Eleventh International Conference,1994,Elsevier, Rutgers University, New Brunswick, NJ, pp. 121-129.

[21] R. Meiri, and J. Zahavi,"Using simulated annealing to optimize the feature selection problem in marketing applications", European Journal of Operational Research, Vol.171. No. 3, pp.842-858, 2006.

[22] V. Kumar, and R.P. Leone,"Measuring the effect of retail store promotions on brand and store substitution", Journal of Marketing Research, Vol.25, No. 2, pp. 178-185,1988.

[23] R.G. Walters,"Assessing the impact of retail price promotions on product substitution, complementary purchase, andinterstore sales displacement", The Journal of Marketing, Vol.55, No. 2, pp. 17-28, 1991.

[24] H.J. Heerde, S. Gupta, and D.R. Wittink,"Is 75% of the sales promotion bump due to brand switching? No, only 33% is", Journal of Marketing Research, Vol.40, No. 4, pp.481–491, 2003.

[25] S. Ma, R. Fildes, and T. Huang. "Demand forecasting with high dimensional data: The case of SKU retail sales forecasting with intra-and inter-category promotional information", European Journal of Operational Research, Vol.249, No. 1, pp.245-2572, 2016.

[26] K. Ord, and R. Fildes,Principles of business forecasting, Mason, 1st ed, OH: South Western Cengage Learning, 2013.

[27] Ö. GürAli, S. Sayın, T. Van Woensel, and J. Fransoo,"SKU demand forecasting in the presence of promotions", Expert Systems with Applications, Vol.36, No. 10, pp.12340-12348, 2009.

[28] P. Baecke, S. De Baets, and K. Vanderheyden,"Investigating the added value of integrating human judgement into statistical demand forecasting systems", International Journal of Production Economics, Vol.191, pp.85-96, 2017.

[29] C. Li, and A. Lim,"greedy aggregation–decomposition method for intermittent demand forecasting in fashion retailing", European Journal of Operational Research, Vol.269, No. 3, pp.860-869, 2018.

[30] N. Kourentzes, and F. Petropoulos,"Forecasting with multivariate temporal aggregation: The case of promotional modelling", International Journal of Production Economics, Vol.181, part A,pp.145-153, 2016.

[31] M.Z. Babai, Y. Dallery, S. Boubaker, andR.Kalai,"A new method to forecast intermittent demand in the presence of inventory obsolescence", International Journal of Production Economics, Vol.209, pp.30-41, 2019.

[32] L. F. Simmons,"Time-series decomposition using the sinusoidal model", International Journal of Forecasting, Vol. 6, No. 4, pp. 485-495, 1990.

[33] L.I. Zheng-Feng, X.U. Guang-Jin, W.A. Jia-Jun, D.U. Guo-Rong, C.A. Wen-Sheng and SH. Xue-Guang, "Outlier Detection for Multivariate Calibration in Near Infrared Spectroscopic Analysis by Model Diagnostics",Chinese Journal of Analytical Chemistry, vol.44, No.2, pp.305-309, 2016 Feb.

Mehdi Farrokhbakht Foumani is an Assistant Professor of Computer Engineering at Islamic Azad University, Fouman and Shaft branch, Iran. He received his Ph.D. in Computer Engineering from Saint Petersburg Electrotechnical University (LETI) in 2013. His research interests include Artificial Intelligence, Machine Learning and Data Mining.

Sajad Moazami Goodarzi Received his M.Sc. in Computer Engineering from the Faculty of Electrical and Computer Engineering at Islamic Azad University, Tehran North Branch in 2018. His research interests include Data Mining and Machine Learning.

A New Power Control Algorithm in MMSE Receiver for D2D Underlying Massive MIMO System

Faezeh Heydari
Department of Electrical Engineering, College of Technical and Engineering, Saveh Branch, Islamic Azad University, Saveh, Iran.
heydari@stu.iau_saveh.ac.ir
Saeed Ghazi-Maghrebi*
Department of Communication, College of Electrical Engineering, Yadegar-e-Imam Khomeini (RAH) Shahre Rey Branch, Islamic Azad University, Tehran, Iran.
s_ghazi2002@yahoo.com
Ali Shahzadi
Department of Telecommunications, Semnan University, Semnan, Iran.
shahzadi@semnan.ac.ir
Mohammad Jalal Rastegar Fatemi
Department of Electrical Engineering, College of Technical and Engineering, Saveh Branch, Islamic Azad University, Saveh, Iran.
j_rastegar@iau-saveh.ac.ir

Received: 02/May/2020 Revised: 18/Oct/2020 Accepted: 06/Jan/2021

Abstract

Device to device (D2D) underlying massive MIMO cellular network is a robust deployment which enables network to enhance its throughput. It also improves services and applications for the proximity-based wireless communication. However, an important challenge in such deployment is mutual interference. Interference, in the uplink spectrum, reusing the same resource with cellular user, is caused by D2D users. In this paper, we study a distributed power control (DPC) algorithm, using minimum mean square error (MMSE) filter in receiver, to mitigate the produced interference in this deployment scenario. For the DPC algorithm, employing the coverage probability of D2D links, an optimal power control approach is proposed, which maximizes the spectral efficiency of D2D links. Using this modeling approach, it is possible to derive closed-form analytical expressions for the coverage probabilities and ergodic spectral efficiency, which give insight into how the various network parameters interact and affect the link. Also, the DPC algorithm is modeled by stochastic geometry and receiver filter is designed by estimation theory that a new structure in this robust network is an approach to improve spectral efficiency. Simulation results illustrate enhancing coverage probability performance of D2D links in term of the target (signal to interference ratio) SIR with respect to different receiver filter and other parameters which are existing in D2D links.

Keywords: Device to Device; Massive MIMO; Power Control; Spectral Eefficiency.

1- Introduction

A. Background

Based on the fifth generation public private partnership (5G-PPP), power consumption is a key performance indicator (KPI) in 5G wireless mobile network. This is declined by many preparations, such as power control (PC) scheme and received filter designed, which result in power efficiency and spectral efficiency enhancement [2]. The power efficiency and spectral efficiency play key role in the 5G structure. In this manner, emerging massive multi input multi output (MIMO) and device to device (D2D) is as a tight key to derive the 5G target such as mobile multimedia, fast mobile internet service, drop data traffic and low latency [3].

In massive MIMO deployment, each base station (BS) is equipped with a significantly great array antennas. As a result, the number of antennas at the BS is much greater than the number of users. Therefore, the link between user and BS will be orthogonal and the process performance in this link becomes simple. It means that in the presence of interference, the system is the same as matched filter [4],[5].

The D2D, without aid of central core and BS, is proximity-based communication in direct transmission [6]. D2D underlying massive MIMO cellular network, with many opportunities and advantages, is a potentially enabled wireless network to improve quality of network performance and to guarantee quality of service (QoS) [7]. However, a main challenge of the mutual interference model is caused by cellular links when sharing the same resource with D2D links. In this paper, we present some methods to alleviate the irritating interference.

The first method is applied to MMSE filter in the receiver with a special structure to boost the desired signal, and also to cancel the interfering signal which influences

* Corresponding Author

the intended signal. It was confirmed that single-stream transmission is favored with respect to multi-stream, when the optimal linear processing strategy, as the MMSE receiver, is employed [8]. Authors in [9] indicated that network throughput can be accelerated linearly with the number of the received antennas, even if a single transmit antenna is merely used at each node. This gain is important specially, to use only linear receive processing such as MMSE and partial zero forcing (PZF). Further, the linear gain is achieved by the usage of maintaining grade of freedom in order to restrain interference and to increase the power of the desired signal simultaneously. Authors in [10], using PZF, studied the spectral efficiency as a metric for D2D underlying massive MIMO communication deployment with perfect and imperfect channel state information (CSI) at receivers.

Next, using an aspect of resource management means PC approach, leads to handle network interference. We investigate the D2D coverage probability $\bar{\mathrm{P}}$ by using the SINR of the uplink for D2D user. In addition, probability should be greater than a minimum SINR such as β. It leads to a good connection in uplink. Then, applying the stochastic geometry tools, the coverage probability is derived [11].

Authors in [11-20], in order to handle the interference and to manage the power resource of network, focused on the D2D underlying cellular networks. To decrease the D2D-to-cellular interference in co-existence networks, in [12] a dynamic PC model was proposed. In [13], for power saving during the transmission, and in the case of the transmit power of D2D users' equipment (DUEs) and cellular users' equipment (CUEs) are greater than a specific value, the powers of these links are matched, since the sum rate of this system is constant. In [14], a centralized PC model, as a solution for managing resource, is proposed for D2D-enable two-tier cellular network. Authors in [15], to decrease the cross-tier interference between the DUEs and CUEs in the uplink communication, studied nominating PC schemes. In this paper, using centralized PC algorithm, two schemes are proposed, and then we use off-on PC algorithm. A centralized PC scheme requires global channel state information, whereas the distributed PC (DPC) scheme requires only direct link information. In [16], a PC algorithm, considering a distributed resource allocation, was verified. Also, using the game theory in uplink system with D2D communication underlying MIMO cellular network, the problem in this model was solved. In [17], for a system with one cell, one D2D pair and multiple CUE, an optimization approach was evaluated to minimize the transmit power.

The authors in [18], to maximize the sum rate in downlink D2D communication with cellular network, introduced a PC solution. It is obvious that PC usually needs the channel state information (CSI). It is difficult in D2D, to set cellular network together in the system. The CSI between links of D2D and cellular communication is a big challenge. This subject was studied in [16-18]. In [19], to obtain the coverage probability and average rate for channel allocation, a D2D overlaid cellular system was studied. The open-loop PC, using a reference certain value as allocated power, was adopted to control CUE transmit power and DUE transmit power [20]. In [21] and [22], to decrease interference in D2D underlying cellular communication, a channel allocation scheme has been studied together with three kinds of PC schemes. The main aspects of these schemes are managing interference, compensating large scale path loss, increasing the sum rate and providing sufficient probability and also, to model this random network to present PC schemes, stochastic geometry as a mathematical technique to describe a system model via special spatial design of user location and to analyze the system interference, was used. Authors in [28] proposed an interference alignment algorithm to reduce the interference of DUE to CUE according to MMSE criterion to derive the received precoding matrix based on the initialized transmitted precoding matrix and then power allocation matrix is derived by using Lagrange duality based PC. Authors in [29] considered pilot-based CSI estimation in which known training sequences are transmitted and used for estimation proposed. During the training phase, different orthogonal set of pilots are used for CUEs and DUEs. Also, the BS uses linear MMSE estimator for the channel estimation. The authors in [30] derived a closed-form spectral efficiency lower bound for the CUE and DUE under impact CSI with maximum-ratio and zero-forcing processing. Also, they used several data PC problems using SE expressions and then, using convex optimization, they solved them. The data PC of CUE and DUE are jointly optimized to guarantee max-min fairness for the cellular and D2D communications. In paper [31], the authors constructed a framework for the joint pilot and data PC for the sum SE maximization in uplink cellular Massive MIMO systems with a variable number of active users. This is a non-convex problem to overcome the inherent non-convexity, an equivalent problem with element-wise convex structure was derived. An alternating optimization algorithm was proposed to find a stationary point.

A channel allocation scheme together with a set of PC schemes in [32] was used to adequately control interference levels under various static and dynamic conditions. The authors used distance-based path-loss parameters (with error margin), varying target SINR in a D2D underlaid cellular system modeled as a random network.

B. Contributions and Outcomes

The main contributions of this paper are summarized as follows. We considered a hybrid network model consist of two kind of ad hoc nodes infrastructure for cellular system and D2D communication. In uplink cellular, users communicate with the BS, while D2D users communicate together with a sufficient distance. The spatial position of D2D and cellular transmitters are modeled by Poisson point process (PPP) with different densities. Such random model, using the stochastic geometry, is used to introduce irregular spatial build of user location. The main challenge of this network is interference between cellular and D2D links. In this paper, to avoid the interference, using MMSE receiver filter before receiver signal and PC algorithm, two approaches are applied. In such network, we propose a PC algorithm which is named distributed algorithm. The main idea of the distributed PC algorithm is to use the CSI knowledge in direct link between transmitter and its corresponding receiver. We derived the coverage probability of D2D links by applying a simple analytic of SINR. Firstly, we compute an approximate expression of ergodic spectral efficiency (ESE) for a typical D2D link. For this purpose, we consider coverage probability of the D2D link. Then, we are able to derive the maximum spectral efficiency of D2D links by achieving the optimal transmission probability. The spectral efficiency of the D2D links is in terms of the density of D2D links and the path loss exponent. In D2D scenario, we suppose that the distance of D2D transmitter to D2D receiver is fixed. However, coverage probability of the cellular link is a function of the distance of CUE from BS. In cellular scenario, the distance of CUE (as transmitter) to BS is variable. Therefore, the optimal transmit power of the CUE will be derived in many different way with respect to distance to the BS.

The remainder of this paper is organized as follows. Section 2 introduces system model with underling massive MIMO in cellular system with designing receiver filter to compute SINR. We present a DPC scheme as an impressive way to decrease the interference in Section 3. The proposed coverage probability to find PC is given in Section 4. In section 5, the spectral efficiency of D2D links is derived by explaining ESE according to DPC scheme. Finally, numerical results and conclusions are shown in Sections 6 and 7 respectively.

2- System Model

In a system model, as shown in Fig. 1, we consider a network uplink which includes a cell with multiple users with one BS in the center of each cell. Also, this BS is equipped with massive MIMO technique and D2D communication underlying. There are two types of users in this system, CUE and DUE which means massive MIMO and underlying D2D respectively. The CUEs and DUEs share the same resource. We assume that the location of the cellular users are modeled as the two dimensions plane according to a homogeneous PPP ϕ_C with density of λ_C. Also, the spatial PPP is based on a uniform distribution of users in the network. The location of D2D transmitter are distributed in homogeneous PPP ϕ_D with density λ_D [25]. We present this system in accordance with single input multi output (SIMO) structure, i.e., a transmitter (either cellular or D2D) applied to an antenna in transmitter. In this case, all receivers have many antennas i.e. the BS has M antennas and D2D receiver has N antennas. If the cell has K cellular users, it should not be bigger than the M. It means that M should be M>>K. In the downlink transmission, the CUE receives signal from the BS and suffers interference from the D2D transmitter caused by sharing the same resource. Also, the D2D receiver suffers the BS interference. Since the power of BS is greater than the D2D receiver, its interference is very annoying. We study network uplink to find a good condition for QoS.

In this system, by proposing the DPC, we follow the scheme to reduce the interference. This model is more considerable than other models, because it only required the CSI of the approach nodes in the direct link. It depends on the distance between the transmitter and receiver. A supportive adaptive way for DPC is proposed, because it supports its link when the SINR reduces from β, then compensates the SINR.

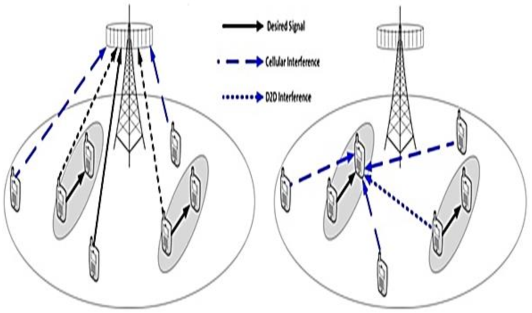


Fig. 1. A signal cell D2D underlying massive MIMO cellular network involving cellular link and D2D links.

The received signals at the BS from typical CUE are as bellow:

$$y_k^{(c)} = \sum_{k \in \phi_M} \sqrt{P_c} \left\| x_k^{(c)} \right\|^{-\alpha_c/2} h_k^{(c)} u_k^{(c)} \\ + \sum_{i \in \phi_D} \sqrt{P_d} \left\| x_i^{(d)} \right\|^{-\alpha_c/2} h_i^{(d)} u_i^{(d)} + n^{(c)} \quad (1)$$

Where $x_k^{(c)}$and $x_i^{(d)}$ denote the distance of cellular transmitter *k* to the BS and the distance of D2D transmitter *i* to the BS, respectively. $\alpha_c > 2$ is the path loss exponent of CUE_BS link. $\boldsymbol{h}_k^{(c)} \in C^{M\times 1}$ and $\boldsymbol{h}_i^{(d)} \in C^{M\times 1}$ are the fading channels from cellular transmitter *k* to the BS and D2D transmitter *i* to the BS, respectively. They are independently distributed with zero mean and unit variance. $u_k^{(c)}$and $u_i^{(d)}$ represent the transmit symbols of cellular transmitter *k* to the BS and D2D transmitter *i* to the BS, respectively. $y_K^{(C)}$ denotes received signal at the BS. $\boldsymbol{n}^{(c)}$ denotes additive white Gaussian noise. Also, $\mathbf{P}_c = [P_{c.0}, \dots P_{c.i} \dots, P_{c.k}]^T$ expresses the profile vector of cellular transmit power with $P_{c.i}$ with the transmit power of the *i*th CUE, and $\mathbf{P}_d = [P_{d.0}, \dots P_{d.j} \dots, P_{d.l}]^T$expresses the profile vector of D2D transmit power with $P_{d.j}$ as the transmit power of the *j*th DUE transmitter. The transmit power constraints are $P_{c.i} \leq P_{max_c}$. $P_{d,i} \leq P_{max_d}$for all links.

In this research, we compute and proof the SINR for typical cellular uplink with the MMSE filter as bellow:

$$SINR_{k.MMSE}^{c} = \frac{P_{k.c}|x_k|^{-\alpha_c}\boldsymbol{h}_k^H\boldsymbol{h}_k}{I_1 + I_2 \ + \sigma^2} \tag{2}$$

$$I_1 = \sum_{i \in \phi_M} P_{i.c}|x_i|^{-\alpha_c}\boldsymbol{h}_i\boldsymbol{h}_i^H$$

$$I_2 = \sum_{j \in \phi_D} P_{j.d}\beta\left|x_j\right|^{-\alpha_c}\boldsymbol{h}_j\boldsymbol{h}_j^H$$

Proof. See Appendix A

Where the numerator represents the desired signal power of CUE K and its denominator represents the summation of interfering signal power of cellular links and D2D links and noise power.

The received signals at D2D receiver from typical DUE are given as

$$\boldsymbol{y}_\ell^{(d)} = \sum_{\ell \in \phi_D} \sqrt{\mathbf{P}_d} \ \left\|x_\ell^{(d)}\right\|^{-\alpha_d/_2} \ \boldsymbol{g}_\ell^{(d)} \ u_\ell^{(d)} + \sum_{j \in \phi_M} \sqrt{\mathbf{P}_c} \ \left\|x_j^{(c)}\right\|^{-\alpha_d/_2} \ \boldsymbol{g}_j^{(c)} \ u_j^{(c)} + \boldsymbol{n}_\ell^{(d)} \tag{3}$$

Where $x_\ell^{(d)}$ and $x_j^{(c)}$ denote the distance of D2D transmitter ℓ to the typical D2D receiver and the distance of cellular transmitter *j* to the typical D2D receiver, respectively. $\alpha_d > 2$ is the path loss exponent of D2D transmitter__ D2D receiver link. $\boldsymbol{g}_\ell^{(d)}$ and $\boldsymbol{g}_j^{(c)}$ are the fading channel from D2D transmitter ℓ to the typical receiver and cellular transmitter *j* to the typical D2D receiver, respectively. $u_\ell^{(d)}$ and $u_j^{(c)}$ represent the transmit symbol of D2D transmitter ℓ to the typical D2D receiver, cellular transmitter *j* to the typical D2D receiver, respectively. $y_\ell^{(d)}$ denotes the received signal at the typical receiver. $\boldsymbol{n}_\ell^{(d)}$ denotes additive white Gaussian noise.

The SINR of typical D2D uplink with MMSE filter is given by

$$SINR_{l.MMSE}^{(d)} = \frac{P_{l.d}|d_{l.l}|^{-\alpha_d}\boldsymbol{g}_l^H\boldsymbol{g}_l}{\sum_{i \in \phi_M} P_{i.c}|d_{l.i}|^{-\alpha_d}\boldsymbol{g}_i\boldsymbol{g}_i^H \ + \sum_{j \in \phi_D} P_{j.d}\left|d_{l.j}\right|^{-\alpha_d}\boldsymbol{g}_j\boldsymbol{g}_j^H + \sigma^2} \tag{4}$$

Where its numerator represents the desired signal power of D2D transceiver pair l and the denominator represents the interfering signal power of cellular links and other D2D links and noise power. In D2D link, we perform the same way as accomplished in cellular link.

3- Distributed Power Control (DPC)

In this section, we present a DPC scheme as an impressive way to decrease the interference. The DPC scheme needs the CSI of direct link between transceiver D2D pair. This scheme needs no transmitter coordinate, since each transmitter can choose transmitter power to maximize its rate from its receiver. It is disregarding any interference and possibly influences its link. This scheme, according to the direct link channel information, chooses power transmitted from the organized set $\{0, P_{max}\}$. P_{max} is assigned to transmitter D2D when the quality of its link is good, i.e.$|\ \boldsymbol{g}_l\boldsymbol{g}_l^H|\ {d_{l.l}}^{-\alpha_d} > \Gamma_{min}$ and 0 assigned in other states. It is formulated as bellow:

$$P_k = \begin{Bmatrix} P_{max,d} & with\ P_s \\ 0 & with\ \overline{P_s} \end{Bmatrix} \tag{5}$$

Where P_s is transmission probability and it is explained as bellow

$$P_s = \left[\mathbb{P}\left[|\ \boldsymbol{g}_l\boldsymbol{g}_l^H|\ {d_{l.l}}^{-\alpha_d} > \Gamma_{min}\right]\right] = \exp(-\Gamma_{min}E[{d_{l.l}}^{\alpha_d}]) \tag{6}$$

In the DPC scheme, each D2D transmitter is chosen its power based on some factors such as gain of its channel $\boldsymbol{g}_l\boldsymbol{g}_l^H$, distance with path loss ${d_{l.l}}^{-\alpha_d}$ and threshold criterion of Γ_{min} . To obtain sufficient PC scheme, we must select a suitable threshold for Γ_{min} . The $\Gamma_{min}(\ P_s)$ plays key role in coverage property and sum rate performance. If Γ_{min} is chosen very big, compare to gain and distance, the transmit probability P_s will be reduced and the interference will be decreased. Decreasing interference leads to decrease the number of active D2D transmitter. These conditions are motivated us to optimize the $\Gamma_{min}(\ P_s)$. Then, we enhance the D2D efficiency in order to optimize the $\Gamma_{min}(\ P_s)$.

Therefore, the appropriate choice for threshold Γ_{min} and consequently P_s can be good modifier for this comparison. So, to culminate the sum rate in this structure, we should optimize the coverage probability.

4- Coverage Probability to Find Power Control

In this section, coverage probability of the D2D link, using stochastic geometry, is computed by concept of mathematical coverage probability. Finally, considering the transmit power limit, an optimal PC algorithm is proposed. In this case, it is assumed that each D2D transmitted power is independent and power distributed function are $F_D(.)$ [34]. However, this algorithm only requires the information in direct link i.e. it is not efficient for cellular communications. So, it does not guarantee the reliability of the whole cellular links. The coverage probability of the cellular links behaves different ways according to the location of the cellular user. It means that the impact of the distance of the uplink user from BS is a key factor to derive the coverage probability [15]. So, the DPC approach is not enough to guarantee reliable cellar links. In this research, we compute and proof the coverage probability of D2D link as bellow

$$\bar{\bar{P}}_{\mathrm{cov}}^{(\mathrm{d})} = \mathbb{P}\left(SINR_{l.MMSE}^{(d)} \geq \beta\right) \tag{7}$$

$$= exp\left(-\frac{\pi\lambda_C(M^{-1}\beta)^{2/\alpha_d}}{\sin c\left(\frac{2}{\alpha_d}\right)}P_{i.c}{}^{2/\alpha_d}P_{l.d}^{-2/\alpha}d_{l.l}{}^2 - \frac{\pi\lambda_D(M^{-1}\beta)^{2/\alpha_d}}{\sin\left(\frac{2}{\alpha_d}\right)}P_{j.d}^{2/\alpha_d}P_{l.d}^{-2/\alpha_d}d_{l.l}{}^2\right)$$

Proof. See Appendix B

5- Spectral Efficiency of D2D Links

The interference of D2D links, that affects *l*-th D2D receiver, is $\sum_{j\in\phi_D} P_{j.d}\beta\left|d_{l.j}\right|^{-\alpha_d}\boldsymbol{g}_j\boldsymbol{g}_j^H$, where $j \in \{1.2.\dots.|\ A|\ \}$. Also, $|\ A|$ is the number of active D2D links that is used in DPC algorithm, i.e.,

$$|\ A| = \lambda\mathbb{P}\left[|\ \boldsymbol{g}_l\boldsymbol{g}_l^H|\ d_{l.l}{}^{-\alpha_d} > \Gamma_{min}\right]\pi R^2 = \tilde{\lambda}\pi R^2 \tag{8}$$

For more explanation, signal transmission from all active link are assumed be Gaussian signal. Therefore, the signals of interference are also Gaussian. The variance of the noise is assumed that $\sigma^2 = 0$, which means SINR is converted to SIR (signal to interference ratio). Consequently, this expression is sufficient to explain ergodic spectral efficiency according to DPC scheme. The spectral efficiency of D2D links is given as bellow

$$R^{(D)} = A \times E[log_2(1 + SIR_l)] \tag{9}$$

$$= A \times \bar{R}_{D2D}$$

$$= \widetilde{\lambda_D}\pi R^2 \times \bar{R}_{D2D}$$

We compute an approximate expression of ESE for a typical D2D link. For this purpose, we consider SIR and coverage probability of the D2D link as bellow

$$\bar{\mathrm{R}}_{\mathrm{D2D}} = \int_0^\infty \log_2(1+\mathrm{x})\left[\mathbb{P}[\mathrm{SIR_l} \geq \mathrm{x}]\right]\mathrm{dx} \tag{10}$$

$$\approx \int_0^\infty \frac{1}{1+x}\,\bar{\bar{\mathrm{P}}}_{\mathrm{cov}}^{(\mathrm{d})}(\mathrm{x})\mathrm{dx}$$

by substituting (7) in (10), we will have

$$= \int_0^\infty \log_2\mathrm{C}\cdot\exp(\mathrm{A}+\mathrm{B})\ \mathrm{dx}$$

Where A, B and C are defined as bellow

$$\mathrm{A} = -\frac{\pi\lambda_\mathrm{C}(\mathrm{M}^{-1}\beta)^{2/\alpha_\mathrm{d}}}{\sin\mathrm{c}\left(\frac{2}{\alpha_\mathrm{d}}\right)}\mathrm{P_{i.c}}{}^{2/\alpha_\mathrm{d}}\mathrm{P}_{\mathrm{l.d}}^{-2/\alpha}\mathrm{d_{l.l}}{}^2$$

$$\mathrm{B} = -\frac{\pi\lambda_\mathrm{D}(\mathrm{M}^{-1}\beta)^{2/\alpha_\mathrm{d}}}{\sin\left(\frac{2}{\alpha_\mathrm{d}}\right)}\mathrm{P}_{\mathrm{j.d}}^{2/\alpha_\mathrm{d}}\mathrm{P}_{\mathrm{l.d}}^{-2/\alpha_\mathrm{d}}\mathrm{d_{l.l}}{}^2$$

$$\mathrm{C} = (1+\mathrm{x})$$

Analytical expression of the typical D2D ESE is obtained by the Laplace transform of link interfering and approximated expression of the uplink interference. However, this is valid for any DPC, when the transmit power is independent from another D2D links. The approximated spectral efficiency of D2D links is given as $R^{(D)}$ that we can maximize it and then we optimize the D2D threshold Γ_{min} , as bellow

$$\mathrm{R^{(D)}} = \widetilde{\lambda_\mathrm{D}}\,\pi\,\mathrm{R}^2\log_2(1+\ \beta)\left[\mathbb{P}[\mathrm{SIR_l} \geq \beta]\right] \tag{11}$$

$$= \lambda_\mathrm{D}\mathrm{P_S}\,\pi\,\mathrm{R}^2\log_2(1+\ \beta)\exp(\mathrm{A}+\mathrm{B})$$

We calculate the optimal transmit power probability, using the optimization problem theory, as bellow.

$$\max \mathrm{R^{(D)}}\ (\beta)$$
$$subject\ to\ \ 0 < P_S < 1 \tag{12}$$

To achieve the optimal transmission probability, we must differentiate even though the objective function is not concave.

$$\frac{\mathrm{d\ R^{(D)}\ (\beta)}}{\mathrm{d\ P_s}} = 0$$

$$1 - \frac{\pi\,\lambda_D\,(M^{-1}\beta)^{2/\alpha_d}}{sinc\,(2/\alpha_d)}\ P_{j\,.\,d}^{2/\alpha_d}\ P_{l\,.d}^{-2/\alpha_d}\ d_{l.l}{}^2\,P_S = 0 \tag{13}$$

To determine the P_s is maximum or minimum, we must use the second order differentiation as bellow

$$\frac{\mathrm{d^2\ R^{(D)}\ (\beta)}}{\mathrm{d\ P_s}} < 0 \tag{14}$$

$$-\frac{\pi\,\lambda_\mathrm{D}\left(\mathrm{M}\,\beta^{2/\alpha_\mathrm{d}}\right)}{\mathrm{sinc}\,(2/\alpha_\mathrm{d})}\ \mathrm{P}_{\mathrm{j\,.\,D}}^{2/\alpha_\mathrm{d}}\ \mathrm{P}_{\mathrm{l\,.d}}^{-2/\alpha_\mathrm{d}}\,\mathrm{d_{l.l}}{}^2 < 0$$

$$P_s = \frac{\sin c\left(\frac{2}{\alpha_d}\right)}{d_{l.l}{}^2\pi\lambda_D(M^{-1}\beta)^{2/\alpha_d}}$$

Then, we have $P_S^* = \min\left\{\frac{\sin c\left(\frac{2}{\alpha_d}\right)}{d_{l.l}{}^2\pi\lambda_D(M^{-1}\beta)^{2/\alpha_d}}\ ,\ 1\right\}$. Now, considering $d_{l.l} = d_0$, the spectral efficiency of D2D links can formulate as bellow

$$\mathrm{R}_{(\beta)}^{(\mathrm{D})} = \begin{cases} \pi\lambda_\mathrm{D}\mathrm{R}^2\mathrm{F}\times\mathrm{E} & \text{for } \beta < \bar{\beta} \\ \mathrm{H}\times\mathrm{G} & \text{for } \beta > \bar{\beta} \end{cases} \tag{15}$$

Where F, E, H and G are defined based on definition of P as bellow:

$$\mathrm{P} = \log_2(1+\beta) \tag{16}$$

$$\mathrm{F} = \mathrm{P}\times\exp\left(-\frac{\pi\lambda_\mathrm{C}(\mathrm{M}^{-1}\beta)^{\frac{2}{\alpha_\mathrm{d}}}}{\sin\mathrm{c}\left(\frac{2}{\alpha_\mathrm{d}}\right)}\left(\frac{\mathrm{P_i,c}}{\mathrm{P_l,d}}\right)^{\frac{2}{\alpha_\mathrm{d}}}\mathrm{d}_0^2\right) \tag{17}$$

$$\mathrm{E} = \exp\left(-\frac{\pi\lambda_\mathrm{D}(\mathrm{M}^{-1}\beta)^{\frac{2}{\alpha_\mathrm{d}}}}{\sin\mathrm{c}\left(\frac{2}{\alpha_\mathrm{d}}\right)}\mathrm{d}_0^2\right) \tag{18}$$

$$\mathrm{H} = \frac{\mathrm{R}^2\sin\mathrm{c}\left(\frac{2}{\alpha_\mathrm{d}}\right)}{\mathrm{d_0}{}^2(\mathrm{M}^{-1}\mathrm{B})^{\frac{2}{\alpha_\mathrm{d}}}} \tag{19}$$

$$\mathrm{G} = \mathrm{P} \times \exp\left(-\frac{\pi\lambda_{\mathrm{C}}(\mathrm{M}^{-1}\beta)^{\frac{2}{\alpha_{\mathrm{d}}}}\,\mathrm{d}_0^2}{\sin\mathrm{c}\left(\frac{2}{\alpha_{\mathrm{d}}}\right)}\left(\frac{\mathrm{P}_{\mathrm{i}},\mathrm{c}}{\mathrm{P}_{\mathrm{l}},\mathrm{d}}\right)^{\frac{2}{\alpha_{\mathrm{d}}}}\right) \tag{20}$$

Where $\bar{\beta} = \left(\frac{\mathrm{sinc}\,({}^{2}/_{\alpha_{\mathrm{d}}})}{\pi\ \lambda_{\mathrm{D}}\left(\mathrm{M}^{-1}\ \beta^{{}^{2}/\alpha_{\mathrm{d}}}\right)}\right)^{{}^{\alpha_{\mathrm{d}}}/_{2}}$.

The spectral efficiency of D2D links is depended on some factors. For example, in the case of $\beta < \bar{\beta}$, the spectral efficiency is derived based on the target SIR value β and density of D2D links λ_D. In the second case, the spectral efficiency is independent of the D2D density and the effect of *M* is lower than the first case. In this research, we compute and proof the spectral efficiency of D2D links by integrating of the spectral efficiency of D2D links in (15) with respect to β. Then, the spectral efficiency of D2D links can be given as

$$\bar{\mathrm{R}}_{(\beta)}^{(\mathrm{D})} = \mathrm{U}+\mathrm{V} \tag{21}$$

Where U and V are defined as

$$\mathrm{U} = \int_0^{\bar{\beta}} \mathrm{U}_1 \mathrm{dx} \tag{22}$$

$$\mathrm{U}_1 = \frac{\mathrm{R}^2 \sin\mathrm{c}\left(\frac{2}{\alpha_{\mathrm{d}}}\right)}{\exp(-1)\mathrm{d_0}^2\left(\frac{\mathrm{x}}{\mathrm{M}}\right)^{\frac{2}{\alpha_{\mathrm{d}}}}}\frac{1}{(1+\mathrm{x})}\exp\left(-\frac{\pi\lambda_{\mathrm{C}}\left(\frac{\mathrm{x}}{\mathrm{M}}\right)^{\frac{2}{\alpha_{\mathrm{d}}}}\mathrm{d}_0^2}{\sin\mathrm{c}\left(\frac{2}{\alpha_{\mathrm{d}}}\right)}(\widetilde{\mathrm{P}})^{\frac{2}{\alpha_{\mathrm{d}}}}\right)$$

$$\mathrm{V} = \int_{\bar{\beta}}^{\infty} \mathrm{V}_1 \mathrm{dx} \tag{23}$$

$$\mathrm{V}_1 = \lambda_{\mathrm{D}}\pi\mathrm{R}^2 \exp\left(-\frac{\pi\lambda_{\mathrm{C}}\left(\frac{\mathrm{x}}{\mathrm{M}}\right)^{\frac{2}{\alpha_{\mathrm{d}}}}}{\sin\mathrm{c}\left(\frac{2}{\alpha_{\mathrm{d}}}\right)}(\widetilde{\mathrm{P}})^{\frac{2}{\alpha_{\mathrm{d}}}}\mathrm{d}_{0)}^2\right)\exp\left(-\frac{\pi\lambda_{\mathrm{D}}\left(\frac{\mathrm{x}}{\mathrm{M}}\right)^{\frac{2}{\alpha_{\mathrm{d}}}}}{\sin\mathrm{c}\left(\frac{2}{\alpha_{\mathrm{d}}}\right)}\mathrm{d}_0^2\right)\frac{1}{(1+\mathrm{x})}$$

It is assumed that distance between transmitter and D2D receiver is fixed and also, DPC scheme in this network operates at maximum power.

6- Numerical Result

In this section, to evaluate the proposed algorithm, numerical results are presented. In this research, the BS is located at the center of the cell with radius of R_c in R^2 plane. The D2D transmitters are distributed according to PPP with density of λ_D. Also, the distance between the transceiver pairs of D2D is *d*. We suppose that the average number of D2D links in the cell equals to $\lambda_D \pi R^2$.

The cellular transmitters have PPP distribution. We applied these results for D2D communication underlying cellular system equipped with two structure, i.e. MMSE filter and PC method. We show how MMSE filter and distributed PC method affect the coverage probability performance gain. Analytical coverage probability performance gain curves for D2D in terms of SIR are achieved. Parameters of the simulated network are shown in Table 1. In this research, we prepared two scenarios. In the first scenario, we compare effects of two filters in present of the DPC on performance gain. In the second scenario, the coverage probability in term of SIR, with MMSE filter and different parameters in D2D links, is derived.

Table 1. Simulation/ Numerical Parameter

Parameter	Value
BS radius of coverage R_c	500m
length of D2D link d	20m
quantity of cellular UEs k	20 to 80
D2D UEs density λ_D	0.00002 and 0.00005
BS antennas quantity M	500
UE Rx antennas quantity N	1and 8
Cellular max Tx power P_c	100mW
D2D max Tx power P_d	0.1mW
Path-Loss exponent of UE-BS $\propto_c$	3.5
Path-Loss exponent of UE-UE $\propto_d$	4

In the first scenario, the coverage probability performance gain of the equipped MMSE filter together with PC algorithm is compared to the case of with PZF filter that the result are shown in Fig. 2. The coverage probability have high amount in lower SIR than higher ones. Also, in the case of using MMSE, performance gain will be enhanced, especially when the target SIR is low. This implies that the desired signal is robust and overcomes the interference. Using MMSE filter in receiver leads to receive the desired signal and overcomes undesired signal. However, the coverage probability of with PZF filter is about 0.2 bps/Hz l less than MMSE filter.

In the second scenario, we compute coverage probability performance with respect to different parameters. Figure 3 illustrates the effects of SIR with variation the number of D2D antennas on the coverage probability of D2D link. As a result, two curves have decreased trend in high *SIR* while the number of antenna is 4 with respect to 8, the coverage probability gain will be increased. The antenna quantity affects the MMSE filter deployment as well as PC case. It is notable, the coverage probability performance gain gradually decreases when the SIR goes up. This is because the SIR is implied interference, so the system interference will be increased as shown in two curves of this figure.

Figure 4 illustrates the effect of SIR with variation of distance between transceiver D2D pair on coverage probability of D2D link. It shows when distance of D2D pair is low, the coverage probability gain in PC algorithm is better than the case of the distance D2D pair is great. For instance, in *SIR*=6 the coverage probability is about 1.99, 1.95, 1.88 in *d*=20, *d*=50, *d*=80 respectively. The significant reason is that maximum power is allocated to transmitter which is at the least distance to its receiver.

Figure 5 shows the effects of SIR with PC and without PC on the coverage probability of D2D links. The coverage probability performance gain is increased by using PC algorithm compared to the case of without PC. Applying PC algorithm leads to mitigate intra D2D interfering and cross-tier D2D and cellular interfering. In the SIR=19, the coverage probability with PC is about 1.82. However, in this SIR we derive coverage probability without PC nearly 1.80.

Figure 6 illustrates the effects of SIR with variation of D2D pair density on the coverage probability of D2D links. This figure demonstrate the analytical result of coverage probability gain and the result of Monte Carlo simulation with the increased rang of SIR and two kind of D2D deployment density. We observe that the analytical performance gain is matched with Monte Carlo simulation. These curves show that D2D coverage probability is improved in high D2D density region in the proposed PC method.

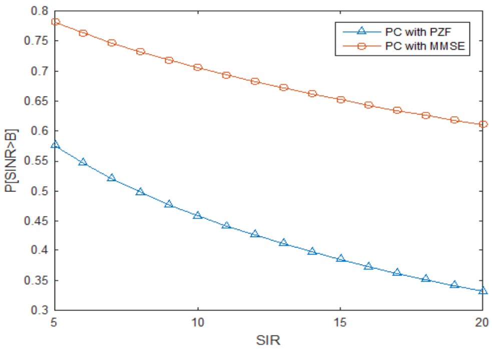


Fig. 2. Coverage probability performance gain of D2D communication with distributed PC algorithm with PZF filter and MMSE filter.

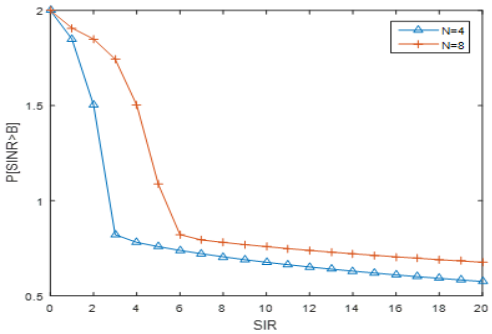


Fig. 3. Coverage probability performance gain of D2D communication with distributed PC algorithm with different number of antenna on D2D transmitter.

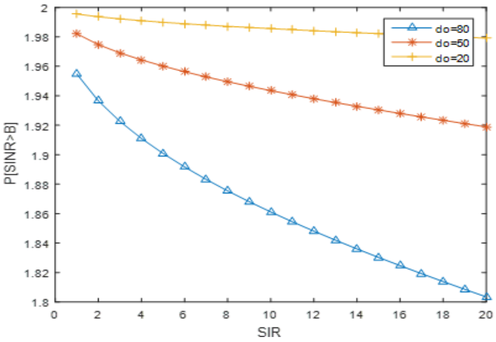


Fig. 4. Coverage probability performance gain of D2D communication with distributed PC algorithm for different distances between transmitter and D2D receiver.

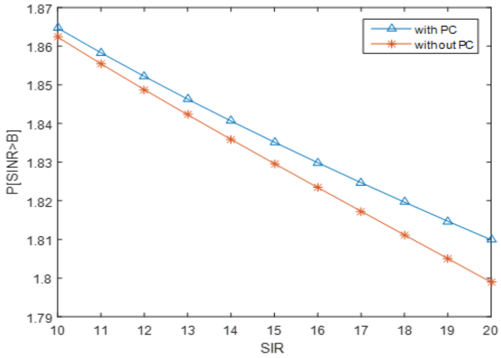


Fig. 5. Coverage probability performance gain of D2D communication according to PC algorithm and the case of without PC.

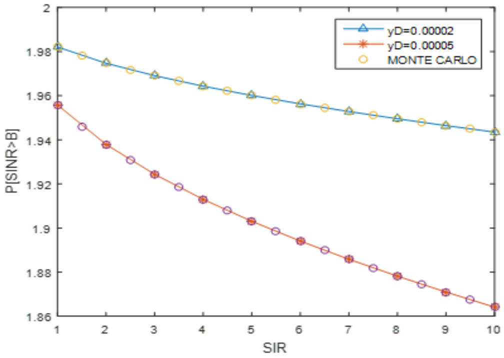


Fig. 6. Coverage probability performance gain of D2D communication with distributed PC algorithm with different densities D2D.

Authors in [20] evaluated spectral efficiency and energy efficiency of the hybrid structure using the power control model. In this research, our final simulation is a comparison between the performances of our proposed method and the used scenario in the article [20]. Figure 7 shows the effects of D2D user density on two scenarios, i.e. the first with the proposed power control model along with the effect of MMSE receiver filter, and the second without using the filter. In our proposed method, the receiver filter model has been added to the proposed system model and we considered power control method. By applying MMSE receiver filter, spectral efficiency calculations with power control and receiver filter are about 4bps / Hz more than spectral efficiency with power control method without receiver filter as considered in [20]. Since we want to show the presence of the receiver filter along with the power control method to improve the simulation curve, we introduced the density of D2D users with very small changes in the surface of a circular cell with radius of 1×10^4m. Thus for both scenarios, without filter as well as with filter the spectral efficiencies are constant curves. Therefore, applying a filter receiver with a special structure in the hybrid network along with the power control model increases the spectral efficiency compared to a mechanism with the same structure without a filter.

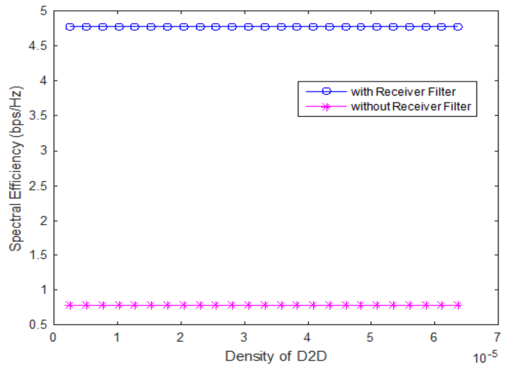


Fig. 7. The effect of D2D users density with receiver filter and without receiver filter on spectral efficiency.

7- Conclusion

We cross check the model consisting of D2D underlying massive MIMO. The network with such deployment deals with challenging interference. Mitigating of interference is the result of enhancing an important metrics such as spectral efficiency.

We followed two way to achieve high spectral efficiency, utilizing MMSE filter after output of channel gain and applying the PC scheme (as a way for managing power resource). They are important approaches to improve network performances. As a result, both of these approaches enable the network to reduce interference. Since MMSE is based on the estimation theory and the model of PC is according to stochastic geometry. We derived the coverage probability of D2D links in the network, eventually spectral efficiency of D2D links has been calculated according to ESE. Also, the spectral efficiencies of D2D links are significantly affect the receiver filter. So, the MMSE filter is the reason of boosting the power of the intended signal and suppressing the interferences. Also, the D2D DPC algorithm is more accommodating to guarantee rate of its links. The DPC method, the optimal PC way is considered, which maximizes the spectral efficiency of D2D links.

APPENDIX

A. Proof of Proposition

The MMSE filter is derived based on the estimation theory, If $s_0$and $\{ s_1 s_2, \dots, s_N\}$ are favored signal and set of interfering signals, respectively. Each signal is iid zero-mean process with magnitude variance a^2 . As where $g_0 = \boldsymbol{w}^H \boldsymbol{h}_0$ and $g_n = \boldsymbol{w}^H \boldsymbol{h}_n$ are the desired signal gain and the *n*th interfering signal gain, respectively, Also $\hat{s}$ is a random signal that is estimated by the desired signal s_0 , and s_n the *n*th interfering signal. $\boldsymbol{h}_1 \boldsymbol{h}_2, \dots, \boldsymbol{h}_N$ are the M × 1 channel vectors and Each vectors is independent zero-mean unit variance complex Gaussian random variable. η_0 is an M × 1 noise vector that is an independent zero-mean complex Gaussian random vector with $\sigma^2 \boldsymbol{I}$ variance. $\boldsymbol{w}^H$ is the M × 1 vector of MMSE filter. the favored signal is estimated by

$$\hat{s}_0 = g_0 s_0 + \sum_{n=1}^{L} g_n s_n + \eta_0 \tag{24}$$

Where $g_0 = \boldsymbol{w}^H \boldsymbol{h}_0$ and $g_n = \boldsymbol{w}^H \boldsymbol{h}_n$ are the desired signal gain and the *n*th interfering signal gain, respectively, Also $\hat{s}$ is a random signal that is estimated by the desired signal s_0 and s_n the *n*th interfering signal [23],[24].

MMSE filter in the receiver leads to minimum mean square error between the desired signal and the output of the receiver, so the SINR will be maximized.

$$\in = \mathrm{E}[|\ \hat{s}_0 - s_0|\ ^2] \tag{25}$$

$$\in = a^2 (W^H A W - W^H h_0 - h_0^H W + 1)$$

$$W = \arg\min \mathrm{E}\ [\hat{s} - s1] \tag{26}$$

Where each signal $\{s_0 . s_1 . \dots . s_L\}$ has zero-mean and a^2 variance. Also,

$$A = (h_0 h_0^H + \sum_{n=1}^{L} h_n h_n^H + \frac{\delta^2}{a^2} I) \tag{27}$$

$$\sum = \sum_{n=1}^{L} h_n h_n^H + \frac{\delta^2}{a^2} I \tag{28}$$

Where $Aw = h_0 . W = A^{-1} h_0$

$$\begin{aligned} &\in = a^2 (1 - h_0^H w) \\ &\quad \in = a^2 (1 - g_0^*) \end{aligned} \tag{29}$$

after some algebra manipulation, the MSE is derived as

$$= a^2 (1 - \boldsymbol{h}_0^H \Sigma^{-1} \boldsymbol{h}_0)^{-1} \tag{30}$$

Based on minimum MSE criterion, the SINR according the MMSE filter is given as bellow

$$\begin{aligned} &\mathrm{SINR} = \frac{\mathrm{E}[|\ g_0 s_0\,|\ ^2]}{\mathrm{E}\left[|\ \sum_{n=1}^{L} g_n s_n + \eta_0\,|\ ^2\right]} \\ &SINR = \boldsymbol{h}_0^H \sum^{-1} \boldsymbol{h}_0 \end{aligned} \tag{31}$$

B. Proof of Theorem

$$\overline{\overline{P}}_{\mathrm{cov}}^{(\mathrm{d})} = \mathbb{P}\left(SINR_{l.MMSE}^{(d)} \geq \beta\right) \tag{32}$$

$$= \mathbb{P}\left(1 \geq \beta \left(\sum_{i \in \phi_M} P_{i.c} |d_{l.i}|^{-\alpha_d} g_i g_i^H + \sum_{j \in \phi_D} P_{j.d} \beta |d_{l.j}|^{-\alpha_d} g_j g_j^H + \sigma^2 \right) P_{l.d}^{-1} {d_{l.l}}^{\alpha_d} (\boldsymbol{g}_l^H \boldsymbol{g}_l)^{-1} \right)$$

It is assumed that x is exponentially distributed with unit mean and unit variance. Then, we will have $\mathbb{P}(\mathrm{X} \geq C) = \exp(-C)$.

$$= E\left(exp\left(-\beta P_{l.d}^{-1} {d_{l.l}}^{\alpha_d} (\boldsymbol{g}_l^H \boldsymbol{g}_l)^{-1} \left(\sum_{i \in \phi_M} P_{i.c} |d_{l.i}|^{-\alpha_d} \boldsymbol{g}_i \boldsymbol{g}_i^H + \sum_{j \in \phi_D} P_{j.d} \beta \left|d_{l.j}\right|^{-\alpha_d} \boldsymbol{g}_j \boldsymbol{g}_j^H \sigma^2 \right)\right)\right) \tag{33}$$

By definition, $g_l^H g_l$ is the square norm of $||g_l||^2$ and also it is defined a χ_{2s}^2 random variable [29].

$$\mathrm{g}_\mathrm{l}^\mathrm{H} \mathrm{g}_\mathrm{l} = ||\mathrm{g}_\mathrm{l}||^2 \sim \chi_{2\mathrm{M}}^2 \tag{34}$$

Therefore, $g_i g_i^H$ is a linear combination of complex Gaussian random variables that equals G_i and thus G_i is iid unit mean exponential. It follow that

$$\mathrm{g}_\mathrm{i} \mathrm{g}_\mathrm{i}^\mathrm{H} \sim \mathrm{Exp}(1) \tag{35}$$

Using Eq. (33) we obtain as below

$$= \mathrm{E}\left(\exp\left(-\gamma\left(\sum_{\mathrm{i} \in \phi_\mathrm{M}} \mathrm{P}_{\mathrm{i.c}} |\mathrm{d}_{\mathrm{l.i}}|^{-\alpha_\mathrm{d}} + \sum_{\mathrm{j} \in \phi_\mathrm{D}} \mathrm{P}_{\mathrm{j.d}} \beta \left|\mathrm{d}_{\mathrm{l.j}}\right|^{-\alpha_\mathrm{d}} + \sigma^2\right)\right)\right) \tag{36}$$

$\gamma = \beta P_{l.d}^{-1} {d_{l.l}}^{\alpha_d} M^{-1}$

Conditioning with respect to transmit power and distance yields as follow

$$\begin{aligned} &= E_{P_{l.d}^{-1} {d_0}^{\alpha_d}}\left(exp(-\gamma\sigma^2)\right) \\ &\times E_{P_{l.d}^{-1} {d_{l.l}}^{\alpha_d}}\left(exp\left(-\gamma \sum_{i \in \phi_M} P_{i.c} |d_{l.i}|^{-\alpha_d}\right)\right) \\ &\times E_{P_{l.d}^{-1} {d_{l.l}}^{\alpha_d}}\left(exp\left(-\gamma \sum_{j \in \phi_D} P_{j.d} \beta \left|d_{l.j}\right|^{-\alpha_d}\right)\right) \end{aligned} \tag{37}$$

Using Laplace transform yields

$$\begin{aligned} &\mathrm{E}\left(\exp\left(-\mathrm{s}\left(\textstyle\sum_{\mathrm{i} \in \phi_\mathrm{M}} \mathrm{P}_{\mathrm{i.c}} |\mathrm{d}_{\mathrm{l.i}}|^{-\alpha_\mathrm{d}}\right)\right)\right) = \\ &\exp(-\frac{\pi}{\sin \mathrm{c}\left(\frac{2}{\alpha_\mathrm{d}}\right)} \mathrm{E}\left({\mathrm{P}_{\mathrm{i.c}}}^{2/\alpha_\mathrm{d}}\right) \lambda_\mathrm{C}(\mathrm{s})^{2/\alpha_\mathrm{d}}) \end{aligned} \tag{38}$$

Applying Laplace transform we can obtain

$$\begin{aligned} &= E_{P_{l.d}^{-1} {d_{l.l}}^{\alpha_d}}\left(exp(-\gamma\sigma^2)\right) \\ &+ exp\left(-\frac{\pi}{\sin c\left(\frac{2}{\alpha_d}\right)} E\left({P_{i.c}}^{2/\alpha_d}\right) \lambda_C(\gamma)^{2/\alpha_d}\right) \\ &+ exp\left(-\frac{\pi}{\sin c\left(\frac{2}{\alpha_d}\right)} E\left({P_{j.d}}^{2/\alpha_d}\right) \lambda_D(\gamma)^{2/\alpha_d}\right) \end{aligned} \tag{39}$$

Deconditioning with corresponding to transmit power and distance yields the coverage probability as follow

$$= exp\left(-\frac{\pi}{\sin c\left(\frac{2}{\alpha_d}\right)} E\left({P_{i.c}}^{2/\alpha_d}\right) \lambda_C(\gamma)^{2/\alpha_d} - \frac{\pi}{\sin c\left(\frac{2}{\alpha_d}\right)} E\left({P_{j.d}}^{2/\alpha_d}\right) \lambda_D(\gamma)^{2/\alpha_d}\right) \tag{40}$$

References

[1] J. G. Andrew, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, “What will 5G be?” IEEE J. Sel. Areas Commun, 2014, vol. 32, no. 6, pp. 1065–1082.

[2] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K. K. Wong, R. Schober, and L. Hanzo, “User association in 5G networks: A survey and an outlook,” IEEE Commun. Surveys & Tutorials, 2016, vol. 18, no. 2, pp. 1018–1044.

[3] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for next generation wireless systems,” IEEE Commun. Mag., 2014 vol. 52, no. 2, pp. 186–195.

[4] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, Fundamentals of Massive MIMO. Cambridge University Press.

[5] E. Björnson, J. Hoydis, and L. Sanguinetti, Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency. Now Publishers, Inc., 2017, vol. 11, no. 3-4.

[6] A. Asadi, Q. Wang, and V. Mancuso, “A survey on device-to-device communication in cellular networks,” Commun. Surveys Tuts., 2014, vol. 16, no. 4, pp. 1801–1819.

[7] S. Shalmashi, E. Björnson, M. Kountouris, K. S. Won, and M. Debbah, “Energy efficiency and sum rate tradeoffs for massive MIMO systems with underlaid device-to-device communications,” EURASIP J. on Wireless Comm. and Netw., 2016, vol., no. 1, pp. 175–193.

[8] R. H. Louie, M. R. McKay, N. Jindal, and I. B. Collings,. Spatial multiplexing with MMSE receivers: Single-stream optimality in ad hoc networks. arXiv preprint arXiv:1003.3056, 2010.

[9] N. Jindal, J. G. Andrews, and S. Weber,. Multi-antenna communication in ad hoc networks: Achieving MIMO gains with SIMO transmission. IEEE Trans. Commun., 2011, 59(2), 529-540.

[10] X. Lin, R. W. Heath, and J. G. Andrews, “The interplay between massive MIMO and underlaid D2D networking,” IEEE Trans. Wireless Commun., 2015, vol. 14, no. 6, pp. 3337–3351.

[11] H. ElSawy, E. Hossain, and M. S. Alouini, “Analytical modeling of mode selection and power control for underlay D2D communication in cellular networks,” IEEE Trans. Commun., 2014, vol. 62, no. 11, pp. 4147–4161.

[12] Gu, S. J. Bae, B.-G. Choi, and M. Y. Chung, “Dynamic power control mechanism for interference coordination of device-to-

device communication in cellular networks," in Proc. IEEE Int. Conf. Ubiquitous Future Netw. (ICUFN), Jun. 2011, pp. 71–75.

[13] S. Ali, H. ElSawy, and M. S. Alouini, "On mode selection and power control for uplink D2D communication in cellular networks," in Proc. IEEE Int. Conf. Commun. Workshop (ICCW), 2015, pp. 620–626.

[14] Y. Huang, A. A. Nasir, S. Durrani, and X. Zhou, "Mode selection, resource allocation, and power control for D2D-enabled two-tier cel- lular network," IEEE Trans. Commun., 2016, vol. 64, no. 8, pp. 3534–3547.

[15] N. Lee, X. Lin, J. G. Andrews, and R. W. Heath, Jr., "Power control for D2D underlaid cellular networks: Modeling, algorithms, and analysis," IEEE J. Sel. Areas Commun.,2015, vol. 33, no. 1, pp. 1–13.

[16] W. Zhong, Y. Fang, S. Jin, K. K. Wong, S. Zhong, and Z. Qian, "Joint resource allocation for device-to-device communications under- laying uplink MIMO cellular networks," IEEE J. Sel. Areas Commun., 2015, vol. 33, no. 1, pp. 41–54.

[17] M. Lin, J. Ouyang, and W.-P. Zhu, "Joint beamforming and power control for device-to-device communications underlaying cellular net- works," IEEE J. Sel. Areas Commun., 2016, vol. 34, no. 1, pp. 138–150.

[18] Z. Yang, N. Huang, H. Xu, Y. Pan, Y. Li, and M. Chen, "Downlink resource allocation and power control for device-to-device communication underlaying cellular networks," IEEE Commun. Lett., 2016, vol. 20, no. 7, pp. 1449–1452.

[19] J. Liu, J. Dai, N. Kato, and N. Ansari, "Optimizing uplink resource allocation for D2D overlaying cellular networks with power control," in Proc. IEEE Global Commun. Conf. (GLOBECOM), Washington, DC, USA, 2016, pp. 1–6.

[20] A. He, L. Wang, Y. Chen, K.-K. Wong, and M. Elkashlan, "Spectral and energy efficiency of uplink D2D underlaid massive MIMO cellular networks," IEEE Trans. Commun., 2017, vol. 65, no. 9, pp. 3780–3793.

[21] A. Abdallah, M. Mansour, and A. Chehab, "A distance-based power control scheme for D2D communications using stochastic geometry," in Proc. IEEE Vehicular Technol. Conf. (VTC), Toronto, ON, Canada, 2017, pp. 1–6.

[22] ——, "Joint channel allocation and power control for D2D communications using stochastic geometry," in IEEE WCNC. Conf. (WCNC), Barcelona, Spain, 2018.

[23] Gao, H., Smith, P. J., & Clark, M. V. Theoretical reliability of MMSE linear diversity combining in Rayleigh-fading additive interference channels. IEEE Trans. Commun., 1998, 46(5), 666-672.

[24] Ma, J., Zhang, Y. J., Su, X., & Yao, Y. On capacity of wireless ad hoc networks with MIMO MMSE receivers. IEEE Trans. Wireless Commun., 2008, 7(12), 5493-5503.

[25] T. D. Novlan, H. S. Dhillon, and J. G. Andrews, "Analytical modeling of uplink cellular networks," IEEE Trans. Wireless Commun., 2013, vol. 12, no. 6, pp. 2669–2679.

[26] K. Hosseini, W. Yu, and R. S. Adve, "Large-scale MIMO versus network MIMO for multicell interference mitigation," IEEE J. Sel. Topics Signal Process., 2014, vol. 8, no. 5, pp. 930–941.

[27] U. Schilcher et al., "Interference functionals in Poisson networks," IEEE Trans. Inf. Theory, 2016, vol. 62, no. 1, pp. 370–383.

[28] Zh. Liu, L. Kuang, T. Hu, "Interference alignment for D2D based on power control and MMSE," in proc. EAI International Conf. on Mobile Multimedia Commun., 2017, pp. 74–82.

[29] H. Xu, W. Xu, Z. Yang, J. Shi, and M. Chen, "Pilot reuse among D2D users in D2D underlaid massive MIMO systems," IEEE Trans. Veh. Technol., 2018, vol. 67, no. 1, pp. 467–482.

[30] A. Ghazanfari, E. Bj¨ornson, and E. G. Larsson, "Optimized power control for Massive MIMO with underlaid D2D communications," IEEE Trans. Commun.,2019, vol. 67, no. 4, pp. 2763–2778.

[31] T. V. Chien, T. N. Canh, E. Björnson, and E. G. Larsson, "Power control in cellular massive MIMO with varying user activity: A deep learning solution," May 2019. [Online]. Available: https://arxiv.org/abs/ 1901.03620.

[32] A. Abdallah, M. M. Mansour, and A. Chehab, ''Power control and channel allocation for D2D underlaid cellular networks,'' IEEE Trans. Commun., 2018, vol. 66, no. 7, pp. 3217–3234.

[33] D. Stoyan, W. Kendall, and J. Mecke, Stochastic Geometry and its Applications, 2nd ed. John Wiley and Sons, 1996.

[34] F. Baccelli and B. Błaszczyszyn, Stochastic Geometry and Wireless Networks. NOW: Foundations and Trends in Networking, 2010.

[35] D. Moltchanov, "Distance distributions in random networks," Ad Hoc Networks, 2012, vol. 10, no. 6, pp. 1146–1166.

Faezeh Heydari received the B.S. degree in Physic from Azad University, Tehran North Branch, Iran in 2004, and M.S. degree in Telecommunication from Azad University, Tehran South Branch, Iran, in 2010. she has received Ph.D degree from Azad University, Saveh Branch, Iran in 2021. Her research interests include Massive MIMO Antennas, Device to Device Networks, Power Control.

Saeed Ghazi-Maghrebi was born in Iran in 1963. He received the B.S. degree in electrical engineering from Kerman University, Iran, in 1988, and the M.S. degree from the Khajeh Nasir-edin- Toosi University of technology, Iran, in 1995. He received the PhD degree in digital communications from Islamic Azad University, Iran in 2010. He is associate professor in communication systems engineering. He has published six books. He has received the grade of the second degree of 30th Khwarizmi International Award (KIA) on Feb. 2017. He is designer of Iranian national data centre. He is a dean of Azad University, Yadegar-e-Imam Khomeini (RAH) Share Rey Branch, Tehran, Iran. His current research interests are digital communication, signal processing and adaptive filtering.

Ali Shahzadi is a full-time Associate-Professor of Electerical and Computer Engineering in the Faculty of engineering at Semnan University, Semnan University, Semnan, Iran. He is conducting research activities in the areas of Wireless Communications Networks, Cognitive Radio, Cooperative Communications.

Mohammad Jalal Rastegar Fatemi works as full-assistant professor in the Faculty of Engineering at Azad University, Saveh Branch, Iran.