In the Name of God

Journal of
Information Systems & Telecommunication

Vol. 10, No.1, January-March 2022, Serial Number 37

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

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
Telecommunication 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)

- Abdelkrim, Moussaoui, University of Bejaia, Bejaia Algeria
- Afsharirad, Majid, Kharazmi University,Tehran, Iran
- 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
- Bayat, Peyman, Islamic Azad University, Rasht , Iran
- Bayanati, mahmonir, West Tehran Islamic Azad University, Tehran, Iran
- Bhushan, Megha, DIT University Dehradun, Uttarakhand, India,
- Deldari, Arash, University of Torbat Heydariyeh, Khorasan Razavi, Iran,
- Ebadati, Omid Mahdi, Kharazmi University,Tehran, Iran
- Eshghi, Mohammad, Shahid Beheshti University, Tehran, Iran
- Eldarabadi, Rahim, Hakim Sabzevari University, Sabzevar, Khorasan Razavi, Iran
- Fathi, Amir, Urmia University, Urmia, Iran
- Farbeh, Hamed, Amirkabir University,Tehran, Iran
- Gampa, V P Chandra Sekhar Yadav, K L University, Guntur, India
- Kheirkhah, Fatemeh, ACECR, Tehran, Iran
- Kolahkaj, Maral, Islamic Azad University, Karaj Branch, Iran
- Mahdieh, omid, University of Zanjan, Zanjan, Iran
- Mahmoodzadeh, Azar, Islamic Azad University, Shiraz, Iran
- Moradi, Gholamreza, Amirkabir University,Tehran, Iran
- Mirzaei, Abbas, Islamic Azad University, Ardabil, Iran
- Nangir, Mahdi, University of Tabriz, Tabriz, Iran
- Navi, Keyvan, Shahid Beheshti University, Tehran, Iran
- Okafor, Kennedy Chinedu,Federal University of Technology, Akure, Nigeria
- Rasi, Habib, University of Technology, Shiraz, Iran
- Ramezani, Reza, Isfahan University, Isfahan, Iran
- Roy, Sahadev, NIT Arunachal Pradesh, India
- Saini, Naveen, Woosong University, Daejeon, Korea
- Sarrafzadeh, Abdolhossein, Unitec Institute of Technology, Auckland, New Zealand
- Selvamuthukumaran S, Annamalai University, Tamil Nadu, India
- Shahbahrami, Asadollah, Guilan University, Guilan, Iran
- Solouk, Vahid, Urmia University of technology, Iran
- Yaghoobi, Kaebeh, Ale Taha Institute of Higher Education, Tehran, Iran
- Zayyani, Hadi, Shiraz University of Technology, Shiraz, Iran

Table of Contents

Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 1-10

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

A Hybrid Machine Learning Approach for Sentiment Analysis of Beauty Products Reviews

Kanika Jindal[1*], Rajni Aron[2]

[1].Lovely Professional University, Punjab, India
[2].SVKM's Narsee Monjee Institute of Management Studies, Mumbai, India

Received: 22 Sep 2020 / Revised: 25 Aug 2021/ Accepted: 25 Sep 2021
DOI: https://doi.org/10.52547/jist.15586.10.37.1

Abstract

Nowadays, social media platforms have become a mirror that imitates opinions and feelings about any specific product or event. These product reviews are capable of enhancing communication among entrepreneurs and their customers. These reviews need to be extracted and analyzed to predict the sentiment polarity, i.e., whether the review is positive or negative. This paper aims to predict the human sentiments expressed for beauty product reviews extracted from Amazon and improve the classification accuracy. The three phases instigated in our work are data pre-processing, feature extraction using the Bag-of-Words (BoW) method, and sentiment classification using Machine Learning (ML) techniques. A Global Optimization-based Neural Network (GONN) is proposed for the sentimental classification. Then an empirical study is conducted to analyze the performance of the proposed GONN and compare it with the other machine learning algorithms, such as Random Forest (RF), Naive Bayes (NB), and Support Vector Machine (SVM). We dig further to cross-validate these techniques by ten folds to evaluate the most accurate classifier. These models have also been investigated on the Precision-Recall (PR) curve to assess and test the best technique. Experimental results demonstrate that the proposed method is the most appropriate method to predict the classification accuracy for our defined dataset. Specifically, we exhibit that our work is adept at training the textual sentiment classifiers better, thereby enhancing the accuracy of sentiment prediction.

Keywords: Sentiment Analysis; Machine Learning; Beauty Products; Feature Extraction; Social Media.

1- Introduction

Sentiment Analysis (SA) is a systematic study of the collection and classification of product reviews on various e-commerce platforms [1]. As the online business has become more popular these days, both sellers and customers are interested in asking and providing feedback on e-commerce platforms simultaneously. These opinions and reviews are a kind of verbal communication that includes personalized suggestions and product ratings. These reviews are a guiding tool for companies to improve their product quality and services. They are very beneficial for consumers to help in making decisions regarding the specific product [2]. Presently, the communication conduct of this digital era's customers has been customized towards the beauty industry that developed as a highly competitive business market [3]. Various social media and e-commerce platforms provide reviews and ratings of different types and brands of cosmetics products to consumers.

Amazon is one of the popular e-commerce platforms that is used to make online purchases. The customers can also provide and review feedbacks regarding any purchase or product available on the website [4]. Although it is very beneficial for consumers and vendors, the increasing number of reviews about a product confuses customers to make the right decision [5]. Therefore, a need arises to analyze these online reviews by classifying them as positive or negative, improving the decision-making process [6]. The customers also tend to express their views in their natural language, so extracting and classifying these language-based reviews using sentiment analysis is necessary. Sentiment analysis is a branch of Natural Language Processing (NLP) that can address the above-discussed problem [7]. Machine Learning techniques are used in sentiment analysis tasks to classify these reviews as positive, negative, and neutral [8]. These trained classifiers are processed to attain reasonable accuracy and require ascertaining the textual data pertinent to the current potentials [9].

This paper presented an empirical study of sentiment classification of textual data using the Bag-of-Words

✉ **Kanika Jindal**
kanikajindal11@gmail.com

technique and implemented three machine learning models. In our study, the unstructured data of beauty product reviews are extracted from Amazon. This work involves three steps, i.e., data pre-processing, feature extraction, and sentiment classification. For this, the unstructured reviews are pre-processed in the first step, and the features are extracted using the Bag-of-Words (BoW) model in the next step. A Global Optimization-based Neural Network (GONN) is proposed for the sentimental classification. Then an empirical study is conducted to analyze the performance of the proposed method with other machine learning classification methods, i.e., Naive Bayes, Random Forest, and Support Vector Machine, and K fold cross-validation is performed to evaluate the accuracy of the system. The other parameters such as precision, recall, and F1 score are also evaluated for all the models. It is concluded that the proposed GONN method outperforms all the other classifiers for the classification of the Amazon beauty products dataset and achieves the best accuracy.

The details of this work will be discussed in the following sections. Section 2 describes a review of related work. Section 3 elaborates on the proposed framework and methodology. Section 4 demonstrates the results and performance evaluation from the experimental work. Finally, section 5 summarizes the conclusion and future works.

2- Related Work

The term 'sentiment analysis' has attained extensive growth and attention in recent years [10]. The primary purpose of this technique is to understand the human emotions expresses in the form of sentiments on social media. It plays a significant role in various organizations concerning education, health, the stock market, and numerous products and services. The research work done in this direction is discussed in this section.

The work [11] is a sentiment analysis approach applied to Twitter data collected from disaster responses. The primary purpose is to understand the needs of the affected people so that rescue responders can help better. For this, the sentiments for the humanitarian reliefs obtained by affected people during and after the disaster are analyzed using machine learning methods. The paper [12] analyses public opinions regarding the demonetization policy implemented by the Indian government on November 8, 2016. The data is collected from Twitter for the two weeks after the policy declaration, and state-based analysis is performed on it. It concluded that almost all the states supported this policy after tackling some minor hindrances for some time. The article [13] is about the application development for cosmetics product reviews gathered from a popular website. It scrutinizes both positive and negative reviews about numerous products using Parts of Speech (PoS) tagger and Naive Bayes classifier. The author endorses using both types of comments in an equal ratio to achieve higher accuracy and efficiency.

The authors of this work [14] proposed a framework in which the support vector machine method and three feature selection methods are used. The dataset comprises 200 reviews extracted from www.amazon.com. All three techniques, i.e., Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Principal Component Analysis (PCA), are compared, and it is concluded that the PSO technique resulted in the best accuracy with SVM. According to the authors [15], the application to automatically analyze the sentiments regarding skincare products can be an effective tool these days. It can be beneficial for both consumers and entrepreneurs. This work has been implemented on a web application to analyze skincare-based tweets by applying data pre-processing and classification methods. The performance results were evaluated to be more than 80%.

This paper [16] is based on stock market forecasting by amalgamating the financial market data with the sentiment features. The data was collected from two financial websites, and machine learning methods SVM is used in this work. The day-of-week effect has been contemplated in this study to improve prediction accuracy. Thus, this approach can help to make better investment decisions in the financial market. This work [17] analyses the cosmetics product reviews written in the Thai language by using the Naive Bayes algorithm. The authors have used various techniques to evaluate the significant phrases, such as cosine similarity, page rank, and Hopfield network algorithm. The paper concludes that the results were not very accurate due to highly unstructured social media data and inadequate management of synonyms. In this paper [18], a framework analyses the laptop reviews based on the product's design, performance, and features. The work consists of three phases, i.e., subjective extraction, calculating the frequency of words, and sentiment classification. It can help people to make effective decisions before buying laptops. The future suggestion is to incorporate the system for other domains. This work [4] has evaluated the textual data by considering the aspect level detection as well as bipolar words for the analysis of sentiments. Amazon data has been pre-processed for extracting information and positive or negative sentiments are generated by utilizing the proposed approach. The future work suggests including other challenges like sarcasm and negation related to sentiment analysis work.

3- Proposed Framework

Sentimental analysis of beauty product reviews in social media is the motivating research in this paper. The influence of social media reviews on beauty products has a positive impact on choosing the right product [3]. But to lead the marketplace, the brands may influence the marketing in the review comments. So, finding the sentiment of the review is the most essential to show the effective review to the consumers. Hence, a hybrid machine learning approach is proposed to effectively predict the sentiment of the social media review on beauty products.

The proposed framework has been segmented into three phases, i.e., data collection and pre-processing, feature extraction using the Bag-of-Words (BoW) method, and sentiment classification using Machine Learning (ML) methods. An empirical analysis of all these techniques has been performed to find the performance evaluation metrics, i.e., accuracy, precision, recall, and F1 score. A brief description of the proposed methodology has been conferred in this section. The process flow of the proposed technique is given in Fig. 1.

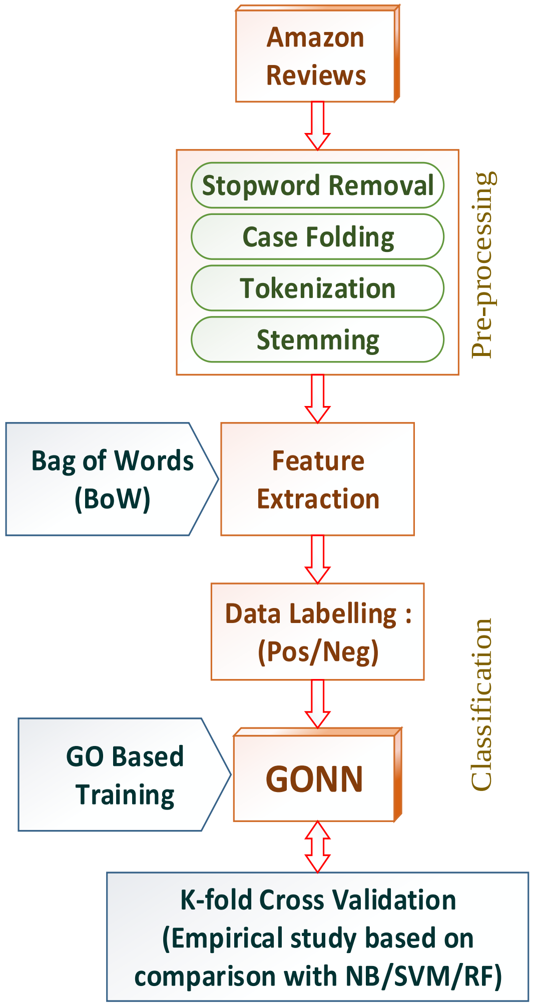


Fig.1 The framework of beauty products review analysis

3-1- Data Collection

The first module involves the collection and pre-processing of data. The dataset used in our work has been accumulated from the gigantic e-commerce platform Amazon.com [19]. It contains an abundant number of reviews based on each product category. This dataset is for various beauty products having 5269 reviews and JavaScript Object Notation (JSON) formats. The various features of each review of the dataset are elucidated in Table 1. An example of the unprocessed dataset is described below.

{"overall": 5.0, "verified": true, "reviewTime": "01 31, 2018","reviewerID": A2IGYO5UYS44RW", "asin": "B00006L9LC", "style": {"Size:": " 281"}, "reviewerName": "Dawna Kern", "reviewText": "I love how soft this makes my skin and the scent is amazing. When my local stored are out I can always get it at Amazon", "summary": "BETTER THAN RAINBATH", "unixReviewTime": 1517356800}

Table 1: Features of the reviews in the dataset

Fields	Description
reviewerID	ID of the reviewer
asin	ID of the product
reviewerName	name of the reviewer
vote	helpful votes of the review
style	a dictionary of the product metadata
reviewText	text of the review
overall	rating of the product
summary	summary of the review
unixReviewTime	time of the review (unix time)
reviewTime	time of the review (raw)

The text and summary of the review and overall rating of the product have been considered for our work from these features. The overall rating contains a rating given by beauty product reviewers. These ratings are expressed in the form of 1-5 stars, with 1 being a bad review and 5 being good reviews. Fig.2 shows the rating distribution of beauty product reviews (1-5 stars). In Fig. 2, the number of reviewers rated the particular review in which the rating 1 is given by 200 reviewers, whereas rating 5 is given by 3750 reviewers.

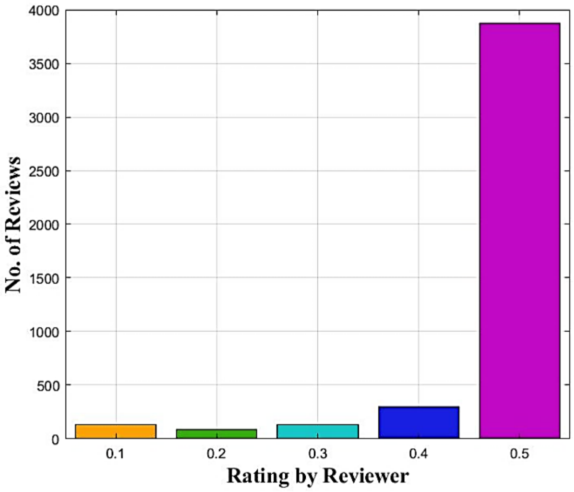


Fig.2 Rating distribution of the reviews in the Amazon dataset

3-2- Data Pre-processing

As the collected data is unstructured and noisy, the entire dataset is pre-processed to form a corpus [20]. The data needs to get clean as much as possible so that the machine learning model can easily understand it and predict whether the review is positive or negative. Therefore, in the data pre-processing process, the entire dataset passes through the following steps:

Stop Word Removal: All the non-relevant words are deleted in this step, like the, and, for, etc. These words do not help predict the polarity of the reviews.

Tokenization: All the relevant words are considered tokens, and all the punctuation marks and special symbols are omitted.

Case-folding: All the tokens are converted into lowercase to avoid repeating the same word in both uppercase and lowercase.

Stemming: Stemming means simplifying each word by its root that indicates enough about what that word means. All the conjugation of the verbs is removed in this step to reduce the redundancy and dimensionality of the sparse matrix.

3-3- Feature Extraction

The Bag-of-Words (BoW) model is used to extract features from the textual reviews collected and pre-processed in the previous phase. These extracted features can be effectively used in machine learning models in the next phase of our methodology. BoW model is used to represent text in the form of a vocabulary that contains the occurrence of the words in the whole document [21]. The frequency of occurrence of each word is assigned a unique number. The features are created by observing all the reviews discretely as an unordered corpus of words to be easily classified afterward. Finally, the textual reviews are fed into machine learning algorithms in the form of numerical vectors.

3-4- GONN -Based Sentiment Classification

The most crucial phase of the proposed work is to evaluate the sentiment prediction accuracy of the reviews expressed by the beauty product users. For this purpose, all the reviews are assigned by the Pos/Neg label to concoct a significant sentiment orientation. The labels are classified depending upon the ratings of the reviews specified by users. This labeled and classified dataset is divided into training (80%) and test (20%) data and implementing machine learning models. Machine Learning methods are best suited for the sentiment classification of these reviews because customers tend to express their suggestions and feedback in their natural language [22].

Hence the GONN is proposed for the effective prediction of the sentiment of the public review. In the proposed GONN, a global optimization technique with a swarm update rule is developed to train the neural network.

3-4-1 Mathematical Modeling of Feed-Forward Neural Network

The proposed neural network consists of three input neurons, one output neuron, and an 'M' hidden neuron. In this model, M is considered as 2. The three input represents three inputs such as word count, character count, and BoW feature. The output neuron represents the class label as positive or negative. The structure of the proposed neural network is given in Fig 3.

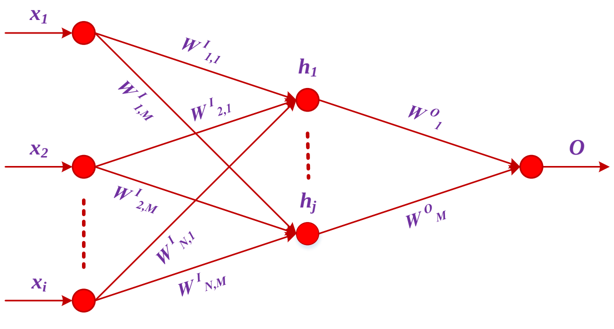


Fig.3 Structure of proposed neural network

Basis function at hidden layer: The basis function calculation is the first step in which the product of input with the weight of the respective link is calculated. The basis function for every node in the hidden layer is calculated as in Eq. (1).

$$b_j = \sum_{i=1}^{N} x_i W_{i,j}^{I} \qquad (1)$$

where ‘b_j’ is the basis function of jth hidden neuron; ‘x_i’ is the ith input value; ‘$W_{i,j}^{I}$’ input weight between ith input neuron and jth hidden neuron, and ‘N’ is the total number of hidden neurons.

Tansig activation function at the hidden layer: The activation function is considered the output of the hidden layer and the input to the output layer. Many functions are available for the activation function calculation, such as tansig, sim, dtansig, logsig. Among them, tansig is the most used and better technique for activation calculation. The activation for every node in the hidden layer is calculated as in Eq. (2).

$$h_j = \left[\frac{2}{1 + exp\left(-2 \times \sum_{i=1}^{N} x_i W_{i,j}^{I}\right)}\right] - 1 \qquad (2)$$

where ‘h_j’ is the activation function of jth hidden neuron.

Neural network output calculation: The output or the obtained output of the proposed neural network is the basis value of the output layer. It is the product of activation value with the respective link in between the hidden and output layer. The output of the neural network is calculated as in Eq. (3).

$$O = \sum_{j=1}^{M} \left\{\left[\frac{2}{1 + exp\left(-2 \times \sum_{i=1}^{N} x_i W_{i,j}^{I}\right)}\right] - 1\right\} W_j^{O} \qquad (3)$$

where ‘O’ is the calculated output of neural network; ‘W_j^{O}’ is the weight between jth hidden neuron and output neuron. Eq. (3) provides the output of the nth training data. After obtaining all the data in the training set, the mean square error (MSE) is calculated as in Eq. (4).

$$Fit = MSE = \frac{1}{T}\sum_{n=1}^{T}(O_n - C_n)^2 \qquad (4)$$

Global optimization based neural network training:
In the conventional neural network, the backpropagation algorithm was widely used for training. Any training algorithm intends to find all the weight values of the network. In a conventional algorithm, a random weight between 0 and 1 would be assigned. Then after calculating the error, its weights are updated. This process is time-consuming and overloading the system. So, the finding of weights value is formulated as an optimization, and global optimization is proposed to find the optimal weight with less mean square error. Hence the accuracy of the system can be improved. The step-by-step procedure of the proposed optimization algorithm is given as follows:

Initialization: In this step, a random set of solutions is generated. The dimension of the solution is the sum of weights required for the proposed model. The range of solution or upper and lower bound of the solution is 0 and 1, respectively. The initial population is represented as in Fig. 4.

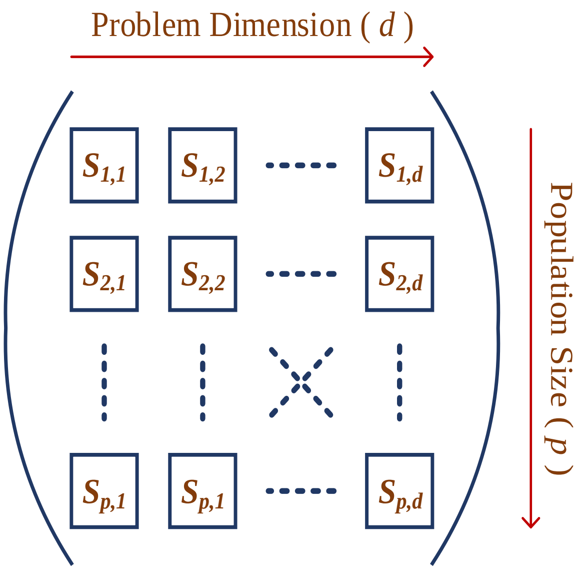


Fig.4 Initial population of proposed global optimization

In Fig. 3, the ‘d x p’ matrix is given, where ‘d’ is the dimension of the problem and ‘p’ is the population size. The population size is random can be any size. The large size of the population consumes execution time and converges at earlier iteration. But the dimension of the population is based on the number or required weights, which can calculate using Eq. (5).

$$d = (N \times M) + M \qquad (5)$$

Fitness Calculation: In this step, the fitness value for every solution set (singe row of the population) is calculated. The objective of this global optimization is to find the optimal weight for the neural network. So, the MSE has given in Eq. (4) is considered to evaluate fitness. The fitness evaluation is utilized to find the current best (C_{best}) and global (G_{best}) best values. The $C_{best}$is the best solution set among the population in the current iteration. The $G_{best}$is the over-best solution obtained among all the iterations as shown in Eq. (6).

$$G_{best}(iter + 1) = \begin{cases} C_{best}(iter) & \begin{cases} iteration == 1 \\ G_{best}(iter) > C_{best}(iter) \end{cases} \\ G_{best}(iter) & otherwise \end{cases} \qquad (6)$$

Update Rule: After fitness evaluation, the solutions are updated based on a swarm rule. The swarm rule used here is referred to from [23].

$$\begin{aligned} pos(iter+1) = w \times pos(iter) \\ + C_1 r_1 \left(C_{best} - sol(iter)\right) \\ + C_2 r_2 \left(G_{best} - sol(iter)\right) \quad (7) \end{aligned}$$

$$sol(iter+1) = sol(iter) + pos(iter+1) \quad (8)$$

In Eq. (7) and Eq. (8), 'pos' is the position value used, which is determined to find the new solution. The 'pos' of iteration 1 ($iter = 1$) is considered as 0, i.e.,$pos(1) = 0$. The parameters 'w, C_1, C_2, r_1, r_2' are probability values consider between 0 and 1.

Termination Criteria: The above steps are repeated for the maximum iteration. If the process meets maximum iteration, then the process is terminated by considering the G_{best} is the best solution or the optimal solution.

3-4-2 Empirical Study to Analyze the Effectiveness

In this empirical study, a comparison-based analysis is performed. Here some conventional machine learning algorithms are considered for comparison. The machine learning algorithms used in our study are Naive Bayes, Random Forest, and Support Vector Machine. The entire dataset is fed into these classifiers, and empirical analysis is performed. After that, K fold cross-validation (K=10) is performed to evaluate the best classifier based on the predicted accuracy attained by the machine learning methods. The overview of our framework for beauty products review analysis has been diagrammatically represented in Fig. 1.

4- Experimental Results and Analysis

The experimented data has been collected from Amazon for beauty product reviews posted by reviewers. Amazon reviewers can provide a product rating from 1 (lowest) to 5 (highest) stars. In our work, the rating stars have been utilized for labeling the reviews. The reviews having 3-star ratings are discarded in our study because this rating is considered neutral (neither positive nor negative) usually.
Therefore, the dataset contains a positive (Pos) label for all those reviews that are 1- or 2- stars and a negative (Neg) label for 4- or 5-stars reviews. Table 2. shows an overview of the product reviews after assessing positive and negative labels based on ratings. The reviews having less than five words are also removed. So, the final pre-processed and labeled dataset, containing 4200 reviews, is being executed by all the three machine learning classifiers, i.e., Naive Bayes, Random Forest, and Support Vector Machine.

Table2: The polarity of the reviews

Review	Sentiment
As advertised. Reasonably priced	Pos
Like the order and the feel when I put it on...	Pos
I bought this to smell nice after I shave it ...	Neg
HEY!! I am an Aqua Veleva Man and abs...	Pos
If you ever want to feel pampered to a sha...	Pos
If you know the secret of Diva you'll LOVE...	Pos
Got this shampoo as a solution for my wife's...	Pos
No change my scalp still itches like crazy...	Neg
Too expensive for such poor quality. Ther...	Neg
It dries my hair doesn't help to reduce dand...	Neg
Outstanding! Tob organic shampoo!	Pos
So watered down I didn't feel like it was a...	Neg
10 stars night here. This product helped me...	Pos
First hair care product I've decided to purc...	Pos
Mad dandruff worse and irritated rest of s...	Neg
Worst shampoo I've ever used. Was mostly...	Neg
Made my hair brittle and dull-looking didn...	Neg
I received the shampoo because I was suff...	Pos

4-1- Evaluation Metrics for Performance Measurement

The evaluation metrics are the fundamental values to evaluate the performance of text classification [24]. The sentiments classified in positive and negative polarity are identified by creating a confusion matrix of true positive, false positive, true negative, and false negative. Accuracy, precision, recall, and F1-score are the significant measures that can be gauged from the confusion matrix based on mathematical rules. The aspects of a confusion matrix are shown below in Table 3.

Table 3: Confusion Matrix

		Predicted Values	
		Positive	Negative
Actual Values	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

The parameters emphasized in the above table are described as:

- True Positive (TP) is the positive value that is correctly identified as positive.
- False Positive (FP) is the negative value that is incorrectly identified as positive.
- False Negative (FN) is the positive value that is incorrectly identified as negative.
- True Negative (TN) is the negative value that is correctly identified as negative.

Precision, recall, F1-score, and accuracy metrics have been computed using the derived values of these parameters. The precision determines the total number of reviews that are accurately classified as positive. Recall determines the total number of reviews that are accurately classified as negative. F1-score measures the weighted harmonic mean of both precision and recall and merges them in a single metric. Accuracy is the simplest metric used to measure the frequency of correct predictions rendered by machine learning models. These metrics are represented by Eq. (6), (7), (8), (9) described below.

$$Precision = \frac{TP}{TP + FP} \qquad (6)$$

$$Recall = \frac{TP}{TP + FN} \qquad (7)$$

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (8)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (9)$$

4-2- K-fold Cross-Validation

K-fold cross-validation is an evaluation procedure used to attain the maximal efficiency of machine learning models [25]. In this work, the cross-validation method divides the dataset into k subsets that are reiterating k times. In every split of data, that kth fold denotes the test data, and the rest k-1 denotes the training data. The machine learning algorithms used in our experimental work are Support Vector Machine (SVM), Random Forest (RF), and Naive Bayes (NB). So, the cross-validation method has been performed on all three models for ten folds (k=10) to attain the best accurate classifier. Table 4 illustrates the accuracy determined by the cross-validation method employed on all three algorithms. The performance of other evaluation metrics, such as precision, recall, and F1-score for NB, SVM, and RF, can also be seen in the table given below.

Table 4: Experimental results of evaluation metrics for all the machine learning methods on the dataset

Methods	**Accuracy 10-fold**	**Precision**	**Recall**	**F1-score**
Naive Bayes	82.96%	97.92%	83.04%	89.87%
Support Vector Machine	95.87%	97.37%	97.86%	97.61%
Random Forest	96.65%	97.14%	98.49%	97.81%
GONN	97.51%	96.07%	98.98%	97.51%

Based on the above table, it is concluded that the proposed GONN, along with Support Vector Machine and Random Forest, both achieve accuracies above 90%. The table shows that Naive Bayes determines the total number of accurately classified reviews as positive with the best precision value of 97.92%. It is found that the Random Forest offers the best F1-score that is used to measure the efficiency of sentiment analysis towards the beauty products dataset. The recall values of GONN are highest as compared to the other methods. The bar graph plotted in Fig.5 depicts that GONN outperforms the other two methods in terms of accuracy, i.e., 97.51%, and the Naïve Bayes model has the lowest predictive accuracy, i.e., 82.96%. The performance of the F1 score has been shown in Fig.6. It concludes that the proposed GONN is the most accurate classifier compared to Random Forest, Support Vector Machine, and Naive Bayes.

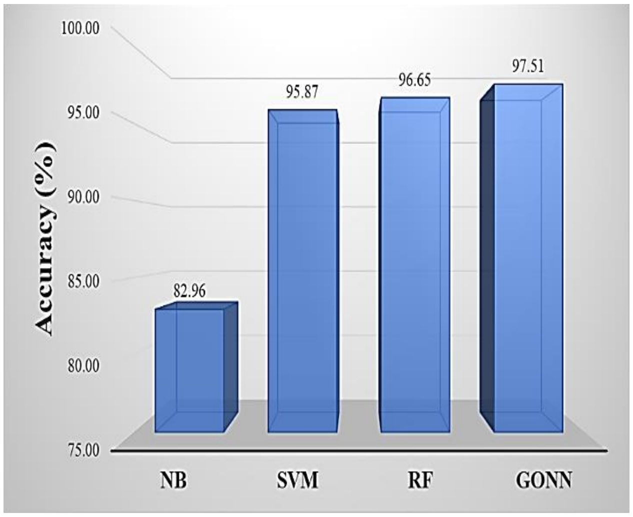


Fig.5 Performance comparison of the techniques regarding the accuracy

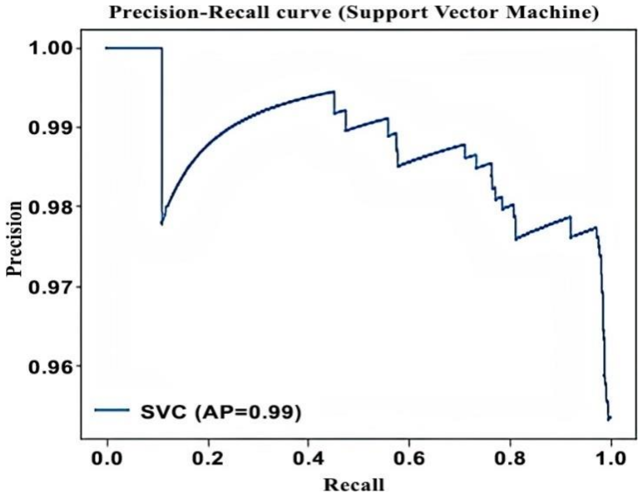


Fig.7 Precision-Recall Curve for Support Vector Machine

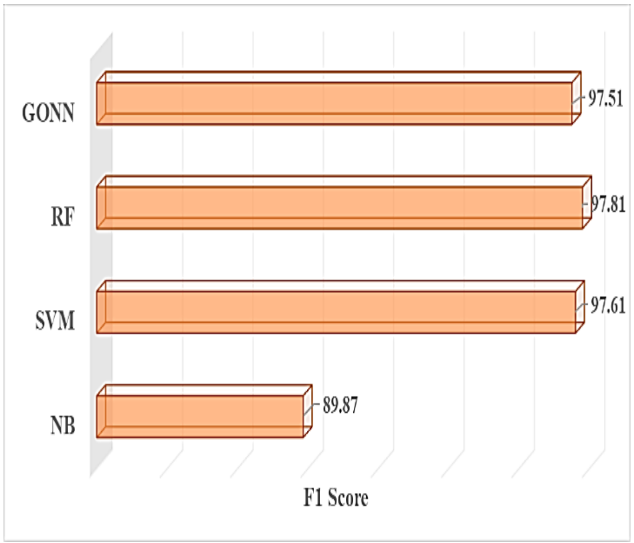


Fig.6 Performance comparison of the techniques regarding F1 score

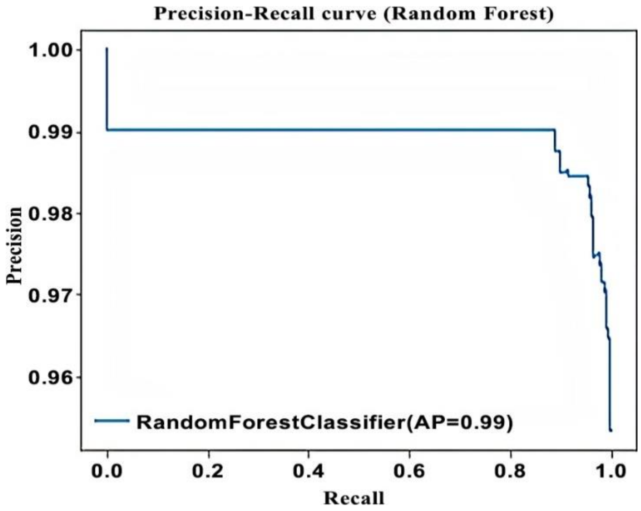


Fig.8 Precision-Recall Curve for Random Forest

Our work has also demonstrated the precision-recall curve for Naive Bayes, Support Vector Machine, and Random Forest diagrammatically. The precision-recall curve is a valuable measure to evaluate the performance of the model [26] visually. It has been effectively used in our work to overcome the limitations of an uneven dataset. The results shown in Fig.7 and Fig.8 illustrates that the average precision of SVM and RF comes out to be comparatively identical. Both are considered good classifiers to predict both the positive and negative classes. Although the average precision of Naive Bayes is slightly less, it is making minor prediction errors among the three methods (Fig. 9).

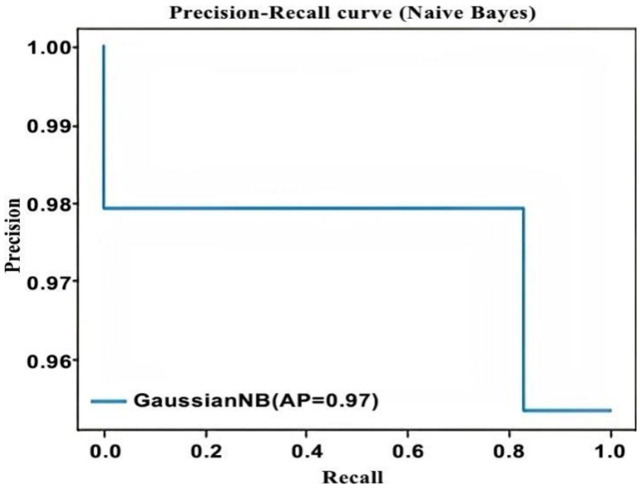


Fig.9 Precision-Recall Curve for Naive Bayes

Table 5: Classification performance comparison

Techniques	Accuracy	Precision	Recall	F1-score
[11]	82.32	84	78	76
[13]	94.17	92	92	92
[14]	83	93	73	81.79
[17]	82.04	77.93	82.4	80.1
[18]	96.36	93.87	95.47	94.66
GONN	97.51	96.07	98.98	97.51

Table 5 shows the comparison of classification performance in terms of accuracy, precision, recall, and F1-score of various techniques in the literature. The comparative analysis in Table 5 clearly shows that the proposed GONN techniques have better performance in all metrics. The best accuracy is achieved by the proposed GONN, which is 97.51%, whereas the second-best accuracy achieved by the technique used in [18] attained 96.36, which is almost 1.15% lesser than the proposed. So, it is evident that the proposed GONN has reasonable due to the effective learning mechanism using global optimization. Similarly, other metrics like precision, recall, and F1-score of proposed GONN is better than the other literary techniques. Based on these performance analyses, it is suggested that the proposed GONN is more suitable for the review analysis than the other techniques.

5- Conclusions and Future Scope

In this paper, we have exhibited the use of machine learning methods to extrapolate the sentiments over the Amazon dataset evoking the opinions and experiences of beauty product users. This empirical work has been carried out using data processing techniques, including stop word and punctuation removal, case folding, stemming, and tokenization in the first phase. Next, the feature extraction process has been implemented by using the Bag-of-Words model. A Global Optimization-based Neural Network (GONN) is proposed for the sentimental classification. Then an empirical study is conducted to analyze the consumers' sentiments towards our dataset by evaluating the performance of the proposed GONN and comparing it with the other machine learning algorithms, such as Random Forest (RF), Naive Bayes (NB), and Support Vector Machine (SVM). These methods categorized the reviews based on positive and negative polarity and cross-validated by ten folds. All the techniques used in our empirical work have been evaluated over the precision, recall, F1-score, and accuracy metrics, and the proposed method has offered the best accuracy results. In future work, we will expand our work by including neutral polarity reviews in the dataset and exploring its effect on the evaluation metrics. Future work will also consider the comparison of other machine learning classification methods and evaluate their performance. The framework implemented in this work will also be adapted for the reviews obtained from other domains.

References

[1] L. Yang, Y. Li, J. Wang and R. Sherratt, "Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning", IEEE Access, vol. 8, pp. 23522-23530, 2020. DOI: 10.1109/access.2020.2969854.

[2] T. U. Haque, N. N. Saber, and F. M. Shah, "Sentiment analysis on large scale Amazon product reviews," *2018 IEEE Int. Conf. Innov. Res. Dev. ICIRD 2018*, no. May, pp. 1–6, 2018, DOI: 10.1109/ICIRD.2018.8376299.

[3] J. Park, "Framework for Sentiment-Driven Evaluation of Customer Satisfaction With Cosmetics Brands", *IEEE Access*, vol. 8, pp. 98526-98538, 2020. DOI: 10.1109/access.2020.2997522.

[4] N. Nandal, R. Tanwar and J. Pruthi, "Machine learning based aspect level sentiment analysis for Amazon products", Spatial Information Research, vol. 28, no. 5, pp. 601-607, 2020. DOI: 10.1007/s41324-020-00320-2.

[5] M. Hu and B. Liu, "Mining and summarizing customer reviews," KDD-2004 - Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 168–177, 2004, DOI: 10.1145/1014052.1014073.

[6] P. Jain, R. Pamula and G. Srivastava, "A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews", Computer Science Review, vol. 41, p. 100413, 2021. DOI: 10.1016/j.cosrev.2021.100413.

[7] X. Fang and J. Zhan, "Sentiment analysis using product review data," J. Big Data, vol. 2, no. 1, 2015, DOI: 10.1186/s40537-015-0015-2.

[8] K. Jindal and R. Aron, "A systematic study of sentiment analysis for social media data", Materials Today: Proceedings, 2021. DOI: 10.1016/j.matpr.2021.01.048.

[9] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," Ain Shams Eng. J.,

vol. 5, no. 4, pp. 1093–1113, 2014, DOI: 10.1016/j.asej.2014.04.011.

[10] Z. Liu, L. Liu, and H. Li, "An Empirical Study of Sentiment Analysis for Chinese Microblogging," Elev. Wuhan Int. Conf. E-bus., 2012.

[11] J. R. Ragini, P. M. R. Anand, and V. Bhaskar, "Big data analytics for disaster response and recovery through sentiment analysis," Int. J. Inf. Manage., vol. 42, no. September 2017, pp. 13–24, 2018, DOI: 10.1016/j.ijinfomgt.2018.05.004.

[12] P. Singh, R. S. Sawhney, and K. S. Kahlon, "Sentiment analysis of demonetization of 500 & 1000 rupee banknotes by Indian government," ICT Express, vol. 4, no. 3, pp. 124–129, 2018, DOI: 10.1016/j.icte.2017.03.001.

[13] P. Pugsee, P. Sombatsri, and R. Juntiwakul, "Satisfactory analysis for cosmetic product review comments," ACM Int. Conf. Proceeding Ser., vol. Part F1287, pp. 0–5, 2017, DOI: 10.1145/3089871.3089890.

[14] D. A. Kristiyanti and M. Wahyudi, "Feature selection based on Genetic algorithm, particle swarm optimization and principal component analysis for opinion mining cosmetic product review," 2017 5th Int. Conf. Cyber IT Serv. Manag. CITSM 2017, 2017, DOI: 10.1109/CITSM.2017.8089278.

[15] P. Pugsee, V. Nussiri, and W. Kittirungruang, Opinion mining for skin care products on twitter, vol. 937. Springer Singapore, 2019.

[16] R. Ren, D. D. Wu, and D. D. Wu, "Forecasting stock market movement direction using sentiment analysis and support vector machine," IEEE Syst. J., vol. 13, no. 1, pp. 760–770, 2019, DOI: 10.1109/JSYST.2018.2794462.

[17] N. Thessrimuang and O. Chaowalit, "Opinion representative of cosmetic products," 20th Int. Comput. Sci. Eng. Conf. Smart Ubiquitos Comput. Knowledge, ICSEC 2016, 2017, DOI: 10.1109/ICSEC.2016.7859945.

[18] T. Chatchaithanawat and P. Pugsee, "A framework for laptop review analysis," ICAICTA 2015 - 2015 Int. Conf. Adv. Informatics Concepts, Theory Appl., 2015, DOI: 10.1109/ICAICTA.2015.7335358.

[19] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 188–197, 2020, DOI: 10.18653/v1/d19-1018.

[20] E. Haddi, X. Liu, and Y. Shi, "The role of text pre-processing in sentiment analysis," Procedia Comput. Sci., vol. 17, pp. 26–32, 2013, DOI: 10.1016/j.procs.2013.05.005.

[21] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: A statistical framework," Int. J. Mach. Learn. Cybern., vol. 1, no. 1–4, pp. 43–52, 2010, DOI: 10.1007/s13042-010-0001-0.

[22] B. K. Bhavitha, A. P. Rodrigues, and N. N. Chiplunkar, "Comparative study of machine learning techniques in sentimental analysis," Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2017, no. Icicct, pp. 216–221, 2017, DOI: 10.1109/ICICCT.2017.7975191.

[23] G. Tomassetti, and L. Cagnina, "Particle swarm algorithms to solve engineering problems: a comparison of performance," Journal of Engineering, vol. 2013, no. 1, pp. 1-13, 2013, DOI: 10.1155/2013/435104.

[24] H. Nguyen, R. Al, and K. Academy, "Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches," SMU Data Sci. Rev., vol. 1, no. 4, 2018.

[25] J. D. Rodríguez, A. Pérez, and J. A. Lozano, "Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 3, pp. 569–575, 2010, DOI: 10.1109/TPAMI.2009.187.

[26] J. Keilwagen, I. Grosse, and J. Grau, "Area under precision-recall curves for weighted and unweighted data," PLoS One, vol. 9, no. 3, pp. 1–13, 2014, DOI: 10.1371/journal.pone.0092209.

Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 11-19

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Remote Monitoring System of Heart Conditions for Elderly Persons with ECG Machine Using IOT Platform

Ngangbam Phalguni Singh[1*],Aditya Kanakamalla[1] Shaik Azhad Shahzad[1], Guntupalli Divya Sai[1], Shruti Suman[1]

[1].Department of Electronics and Communication Engineering ,Koneru Lakshmaiah Education Fondation ,KL Deemed to be Vaddeswaram ,India

Received: 31 Jan 2021 / Revised: 25 Sep 2021/ Accepted: 25 Oct 2021
DOI: https://doi.org/10.52547/jist.15692.10.37.11

Abstract

These days, heart illnesses are viewed as the essential purposes behind unforeseen passing. Along these lines, different clinical gadgets have been created by designers to analyze and examine different infections. Clinical consideration has gotten one of the main issues for the two individuals and government considering enthusiastic advancement in human people and clinical use. Numerous patients experience the ill effects of heart issues making some basic dangers their life, consequently they need ceaseless observing by a conventional checking framework for example, Electrocardiographic (ECG) which is the main procedure utilized in estimating the electrical movement of the heart, this method is accessible just in the emergency clinic which is exorbitant and far for distant patients. The improvement of far-off advancements enables to develop an association of related devices by methods for the web. The proposed ECG checking framework comprises of AD8382 ECG sensor to peruse patient's information, Arduino Uno, ESP8266 Wi-Fi module, and site page. The usage of the proposed ECG medical care framework empowers the specialist to screen the patient's distantly utilizing IoT http application library utilized in Arduino ide compiler to such an extent that it can send that information to website page made, on imagining the patient's ECG signal without human presence site page itself can book arrangement for that persistent, if it is anomalous. The observing cycle should be possible at whenever and anyplace without the requirement for the emergency clinic.

Keywords: IOT; AD8232; Arduino ide; ESP8266; ECG.

1- Introduction

The heart ailments are the basic clarification behind the unexpected passing and causes a high mortality around the world yearly. Amazingly, heart diseases patients are resolved to stay at their home without making any potential that pressure the heart. Nevertheless, an enormous part of the patients fails miserably before they get any treatment since they do not feel crippled until the ailment become at an essential stage. Along these lines, the doctor must screen the actual status of the patient's heart consistently to forestall any dire backslide in heart status and afterward diminishing the passing [1]. It is difficult to keep all heart patients in the facility to screen their status. All things being equal, growing continuous checking medical care framework dependent on remote innovation is a compelling answer for the doctor to screen their patients distantly.

The Electrocardiogram is physiological wave signal that depicts the electrical development of the heart. It is produced because of siphoning and pressing blood between the atria what is more, chamber inside the heart [2]. ECG enables the specialists to choose the clinical state of the heart by assessing the time needed for the electrical sign to be experienced the heart. ECG signal is estimated utilizing a set of cathodes that are put non-carefully on the body surface to acquire data about the electrical action delivered by the heart. The expression "Web of Things" IoT was utilized for the first time in 1999 by Kevin Ashton during his works at MIT's Media Centre. This idea is utilized to communicate the association between the machines and PCs through sensors and utilizing the web as an intend to acknowledge control orders and detailing status [3]. IoT is around for a long time however without a name, machine-to-machine (M2M) was accessible for a long time and is the closest idea to IoT.

✉ **Ngangbam Phalguni Singh**
phalsingh@gmail.com

There are numerous different names for IoT, for example, Ubiquitous figuring and web of everything. For the most part, the term (IoT) alludes to a dynamic and worldwide organization foundation which comprises of an enormous number of organization associations and processing abilities reach out to objects, sensors and ordinary things not ordinarily considered PCs [4]. These devices are allowed to create, eat up, and exchange data with a little human intervention. The gigantic degree use of IoT changes various points to the things, for instance, home computerization, energy the board contraptions, orchestrated vehicles, vigilant traffic besides, prosperity noticing devices. IoT carries inconceivable solace to the clinical administrations field, especially for persevering noticing and following organization. Enlivened improvement of the web disseminated processing and Internet compromise of clinical noticing and the board stage gives new events to the crisis centres and care centres to improve access and interconnection of devices used in clinical administrations [5]. Installed innovations have a significant part in conveying medical care to individuals in isolated and far areas by giving them a checking framework which presents a persistent stream of precise information for better medical services choices. As the innovation is to gather, examine and send information, IoT keeps on improving the IoT-driven medical services applications and frameworks arise. This paper used the IoT to introduce a financially savvy, distantly utilized, effectively prepared, and continuous medical services framework for checking the ECG of the heart patients utilizing ECG sensor and Node MCU. This framework gives ideal answers for heart patients that are living in far off areas and having restricted pay. The remainder of the paper is coordinated as follow [6]. Segment two centres around the connected works, and Section three depicts the framework from three angles: the framework segments, its information obtaining, information preparing, transmission and information representation parts. In Section Four, we present the execution steps and result perception. Segment five talks about the acquired outcomes and examination with a few related works. Segment Six gives the ends and future works.

2- Related Works

G. Xu [6], The rising Internet of Things (IoT) framework permits us to layout small gadgets which could be ready to sense, processing and communicating, permitting sensors, embedding gadgets and other ' things ' to be created in order that you'll assist to apprehend the environment. during this paper, the IoT assisted electrocardiogram (ECG) tracking framework with stable records transmission has been proposed for non-stop cardiovascular fitness tracking. the development and implementation of a light-weight ECG Signal Strength Analysis has been proposed for computerized category and realtime implementation, the utilization of ECG sensors, Arduino, Android phones, Bluetooth and cloud servers with the proposed IoT-assisted ECG tracking system. For stable records transmission, the Lightweight Secure IoT (LS-IoT) and light-weight Access Control (LAC) has been proposed. The ECG alerts taken from the MIT-BIH and Physio Net Challenges databases and ECG alerts for varied bodily sports are analyzed and checked in real-time. The proposed IoT assisted ECG tracking framework has extraordinary capability to make a decision the medical attractiveness of ECG alerts to reinforce the efficiency, accuracy and reliability of an unsupervised diagnostic system.

B. M. Lee and J. Ouyang planned a wise assistance model for medical care [7]. A joint effort convention has been proposed to send and get the elements identified with the dangers between IoT medical care gadgets. The coordinated effort convention is an application convention comprising of numerous occasions which are applied to arrange the progression of information between IoT's gadgets, the boot occasion instates all the IoT's gadgets in the framework by communicating join message.

G. Ismaeel and E. K. Jabar expected to arrive at m-wellbeing by planning a medical care framework for pregnant ladies utilizing Mobile GIS [8]. This framework empowers the pregnant lady which needs exhortation, update and supporting (from their home) to enlist in maternity care focus through the web interface by sending SMS to the framework worker including her ID, telephone number, name, age, just as finding the position utilizing a versatile underlying GPRS method.

B. Padmavathi and S. T. Rana planned and actualized a system for IoT put together medical services arrangement based with respect to distributed computing [9]. This system incorporates five layers. Information preparing layer recognizes and finds the information from sensor-based innovation by utilizing RFID, ZigBee, NFC, Barcode advances and computerized cameras.

A. Ahamed et al. executed a minimal effort ECG observing framework [10]. This framework comprises of four units, an ECG securing unit detects the patient's ECG information signal by utilizing the Bio Protech T716 terminal, this information is moved to the intensifier input. Signal molding unit utilizes the AD620 intensifier just as numerous different methods to get an unadulterated information signal by enhancing the information sign and diminishing the clamor.

The ECG records the electrical activity generated by the depolarization of the heart muscle, which propagates it in response to electrical waves reaching the skin. Although the amount of electricity is actually very small, it can be taken reliably with ECG electrodes connected to the skin. The three major types of ECG are:

2-1- Resting ECG

You lie down, no move is allowed during the test, as electrical impulses developed by other muscles may intrude with those generated by your heart. This usually takes 5 to 10 minutes.

2-2- Ambulatory ECG

If you have an ambulatory, you wear a portable recording device for at least 24 hours. You are free to move around normally while the monitor is attached. This type of ECG is used for people whose syndromes are intermittent and may not show up on a resting ECG, and for people recovering from heart attack to ensure that their heart is functioning properly. You record your syndromes in a note and note when they occur so that your own experience can be compared with the ECG.

2-3- Cardiac stress test

This test is used to record your ECG while you ride on an exercise walk on a treadmill. This type of ECG takes about 10 to 25 minutes to complete [10].

As per the standard assignment from Figure 1, the ECG arrangement comprises of in any event four anodes on the chest or at the four closures. In any case, there are varieties of this setting to take into consideration more adaptable and meddlesome accounts. Terminals are joined to the wrists and lower legs. ECG anodes ordinarily require wet sensors and a conductive gel to build the conductivity between the skin and the cathodes [11].

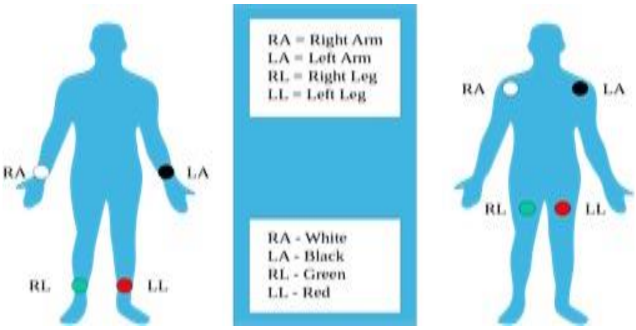


Fig. 1 Says that standard nomenclature (RA = right arm; LA = left arm; RL = right leg; LL = left leg).

Typical range is 120 - 200 ms. Figure 2 shows the QRS length up to 120 ms. The abnormal value of the heart rate is 60 to 100 beats / min. Indicates the heart rate is slower than 60 beats / min, which is called bradycardia. A heart rate of more than 100 beats / min is called a tachycardia [13].

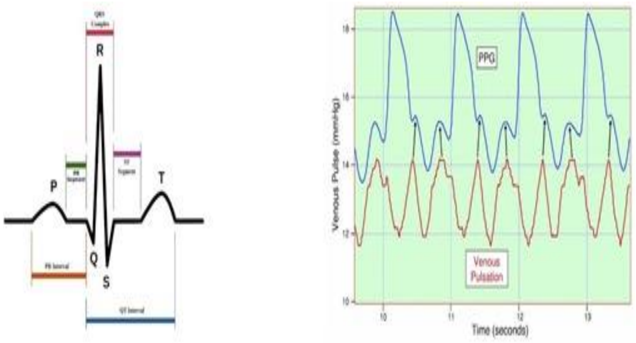


Fig. 2 Measured from first deflection of QRS complex to end of QRS complex at isoelectric line

In rundown, the connected works talked about already do not present an ECG medical care framework that has the four fundamental highlights: continuous, minimal effort, simple prepared and distantly arrangement situated utilized. Any medical care with these four highlights will be fulfilled for the low-pay and distant living heart patients. The proposed medical services framework helps the doctors to follow the actual status of their patients at whenever and anyplace.

3- Materials and Methods

The proposed framework comprises of a bunch of equipment and programming parts that are cooperated to construct the general framework as shown in figure 3. These parts can be partitioned into four units, the information gathering unit that is liable for perusing heart action signals from the human body, the got information from ECG test, ESP Wi-Fi module that sends the created data to the distant page, lastly, investigate the readings and continue as indicated by it [14].

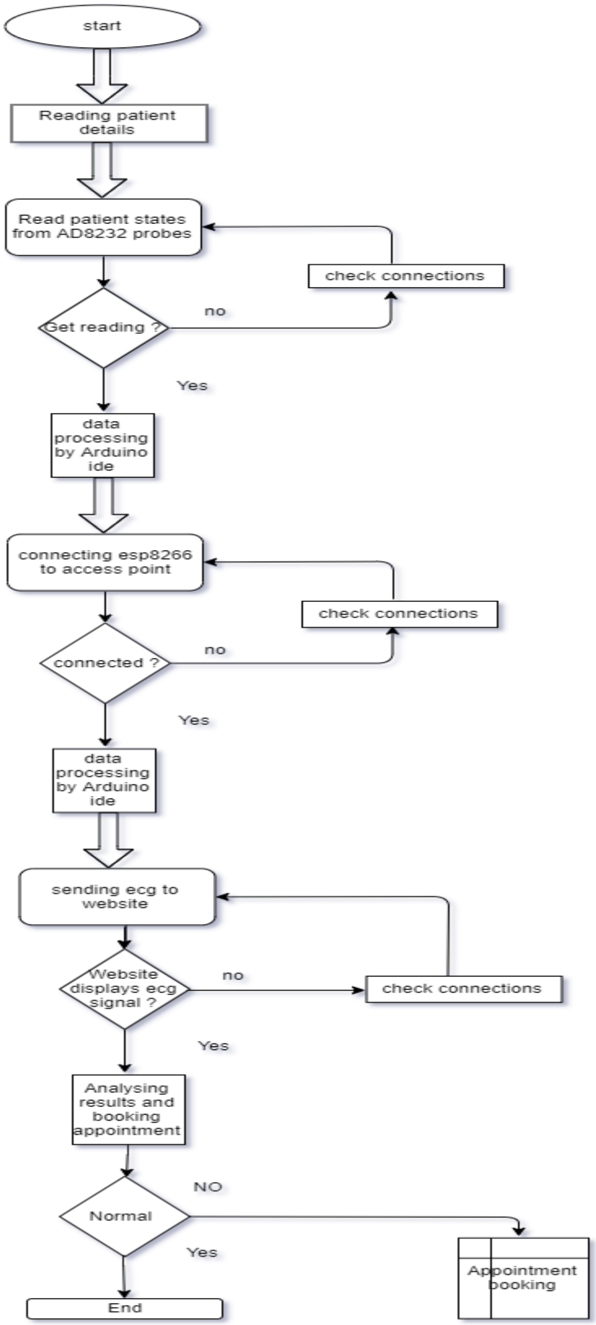

start
Reading patient details
Read patient states from AD8232 probes
check connections
Get reading ?
no
Yes
data processing by Arduino ide
connecting esp8266 to access point
check connections
connected ?
no
Yes
data processing by Arduino ide
sending ecg to website
Website displays ecg signal ?
no
check connections
Yes
Analysing results and booking appointment
NO
Normal
Yes
Appointment booking
End

Fig. 3 Flowchart showing framework

The major pieces of the data making sure about unit are AD8232 chip and three paste cathodes terminals pointed as R for the right side, L for the left side and COM as shoreward the key endeavor of the AD8232 chip is removing, upgrading, and filtering the little signals of the heart electrical activity when a boisterous condition happens in view of development or distant anode plan. Signal trim is cultivated using a bipolar point high pass channel which is joined with the instrument speaker structure.

The AD8232 chip contains a non-operational speaker that makes a three-post low pass channel similarly as decreasing the additional uproar. As shown in figure 4, AD8232 contains a brisk recovery work, we can diminish the long high leave channel to work behind the long tail wonder. If the enhancer rail voltage signal change, (for instance, the lead out of the case), AD8232 worked in an enhancer for the right-side driver (RLD) and other drive lead applications [15]. AD8232 will thus adjust to a higher direct cut-off state.

This incorporate license the AD8232 to achieve quick recovery, and accordingly the lead relationship with the assessment of the article can be cultivated at the soonest opportunity after the assessment of the value. AD8232 is related with the esp8266. The pin specifications are shown in Table 1.

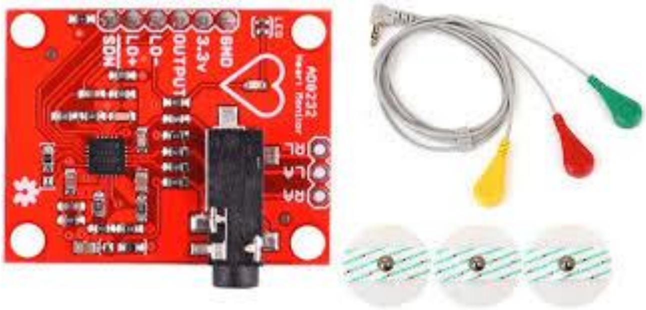


Fig. 4 AD8232 ECG sensor.

Table 1 Pins specifications of AD8232

Board label	***Pin function***	***Node MCU connection (figure 5)***
GND	Ground	GND
3.3V	3.3V power supply	3.3V
OUTPUT	Output signal	A0
LO-	LEADS-OFF detect-	D3
LO+	LEADS-OFF detect+	D2
SDN	Shut down	Not used

Fig 5: Node MCU

The ESP8266 Wi-Fi module which is utilized to send the information from an Arduino from which it can access by the client. Wi-Fi module integrated with TCP/IP protocol and has self-contained SOC. ESP 8266 module can easily provide Wi-Fi when it gets attached to an Arduino [16] – [18].

4- System Implementation

In this segment, the segments of the proposed framework are associated, as shown in figure 6. The execution starts with instating site tolerant subtleties, the initial step is to fill structure, the following stage is to make a record utilizing a substantial email address to get a validation code information field where the information can be shown as a chart so variety of the ECG information sign of the patients can be envisioned. The product is intended for playing out this undertaking.

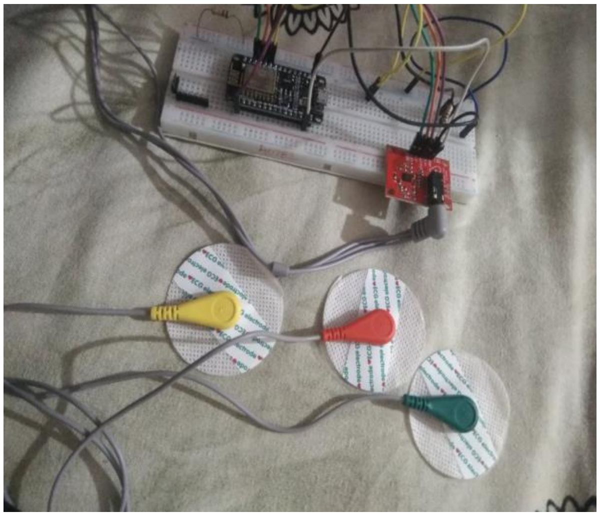


Fig. 6 Proposed Work

The program code is composed utilizing Arduino IDE 1.8.5 climate, this code beginning to set up the essential library for ESP8266 module and web application. Figure 7 shows the circuit diagram of the Arduino connections. The ECG simple signs from the test's terminals are perused Node MCU A0. The test terminals must be set appropriately at the predetermined piece of the human body the information by changing the simple information over to advanced data utilizing Analog to computerized transformation. The yield information will be passed to the ESP8266 module through pin 2 and pin 3. ESP8266 Module is associated with the web hotspot utilizing the name of the hotspot (SSID) and (PASSWORD) and afterward sending the information to site page stage.

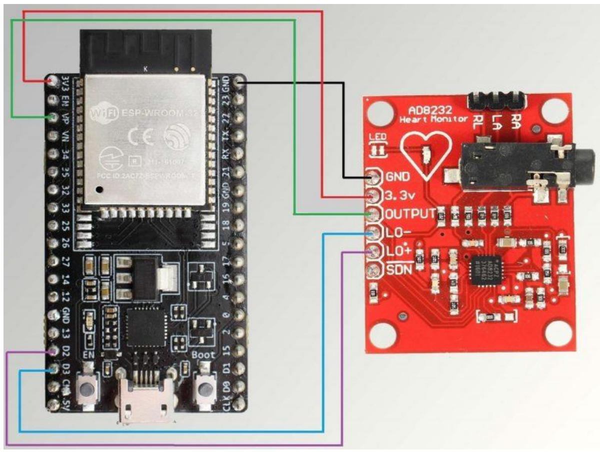


Fig. 7 Circuit Diagram

5- Results and discussions

The basic point of the proposed framework is limiting the time between the snapshot of getting patient's information and showing the outcomes through application on the PC. In this section, the pieces of the proposed structure are related. The execution begins with presenting page, the underlying advance is to sign in to point of arrival or make a record using a real email address where the data can be appeared as a chart and plain. With the objective that assortment of the ECG data indication of the patients can be envisioned. The website page is intended for playing out this errand. Figure 8 shows the data from the monitor of the system of the proposed framework segments.

We have,
(Inter beat internal let take) x (ml in one minute)
= 60*1000 m/s = 60000 m/s

IBI=60000/x BPM
BPM=60000/inter beat internal

Table 2 shows the ECG data generated from the proposed device and table 3 shows the comparison of the proposed work with the related works.

The program code is composed utilizing Arduino IDE climate, this code beginning to set up the fundamental library for ESP8266. The ECG simple signs from the test's terminals are perused Node MCU pin A0. The test terminals must be put appropriately at the predetermined piece of the human body.

Node MCU measures the information by changing the simple information over to advanced data utilizing Simple to modernized change. The yield data will be passed to the ESP8266 module through pin D2 and pin D3. ESP8266 Module is associated with the web hotspot utilizing the name of the Wi-Fi (SSID) and (PASSWORD) and afterward sending the information to the page.

Table 2 ECG data

Interval	***Corresponds to***	***Normal range***
PR interval	AV nodal delay	0.12-0.20 seconds
QRS duration	Ventricular depolarization	Up to 0.10 sec
QT intervals	Total duration of ventricular depolarization	Up to 0.43 sec
R-R intervals	Time between beats -is used to calculate heart rate	0.6 to 1 sec
Mean electrical axis	Net vector of ventricular depolarization	-30-+100 degree is the widest normal range

Table 3(a) Comparing the Proposed System with Related works

Parameter	***Proposed System***	***Ref. [6]***	***Ref. [3], [16]***
Data Reading	AD8232 ECG sensor	ECG sensor	AD8232 ECG sensor
Microcontroller	ESP8266	Arduino, Android phones	Arduino Uno
Transmission Technique	Wi-Fi	Bluetooth	Wi-Fi
Distance	Unlimited	10 meters	Unlimited
Displaying Results	Webpage	webpage	Blynk Application on smartphone
IoT	Yes	yes	Yes
Time	Real-time	Real-time	Real-time
Cost Effective	Yes	yes	Yes

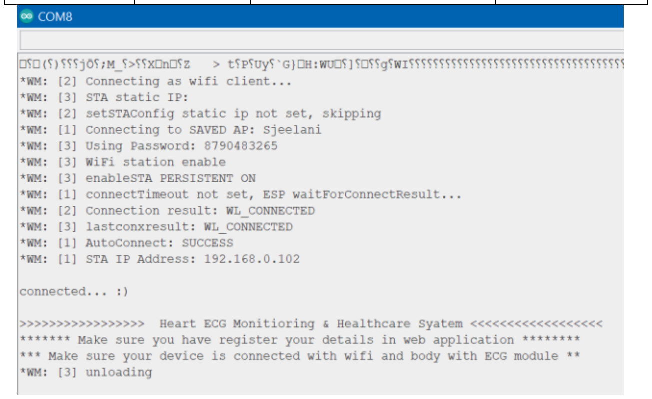
Table 3(b) Comparing the Proposed System with Related works

Parameter	***Ref. [17]***	***Ref. [19]***	***Ref [14]***
Data Reading	Heartbeat pulse, blood pressure and heart sound sensors kits	Heart Beats sensor	AD8232 CG sensor
Microcontroller	Raspberry Pi	Raspberry Pi	Arduino Uno
Transmission Technique	Wi-Fi	GSM+Wi-Fi	Bluetooth
Distance	Unlimited	Unlimited	10 meters
Displaying Results	Web Page IP address	SMS Acknowledgment and Web page IP	Serial plotter and smartphone
IoT	Yes	Yes	No
Time	Real-time	Real-Time	Real-time
Cost Effective	Yes	Yes	Yes

Table 4 A comparison of Arduino, Raspberry Pi and ESP8266 Node MCU

	Arduino UNO	***Raspberry Pi***	***ESP8266 Node MCU***
Developer	Arduino	Raspberry Pi Foundation	ESP8266 open source community
Operating System	None	Linux	XTOS
CPU	Atmel, ARM, Intel	ARM Cortex	LXT106
Clock Speed	16 MHz	1.2GHz	26 MHz – 52 MHz
Memory	32KB	1-4GB	Upto 128MB
Storage	1KB	MicroSDHC Slot	4MB
Power	USB, Battery, Power Supply	USB, Power Supply	USB
Operating Voltage	5V	5V	3.3V

I/O Connectivity	SPI I2C UART GPIO	SPI DSI UART SDIOCSI GPIO	UART, GPIO

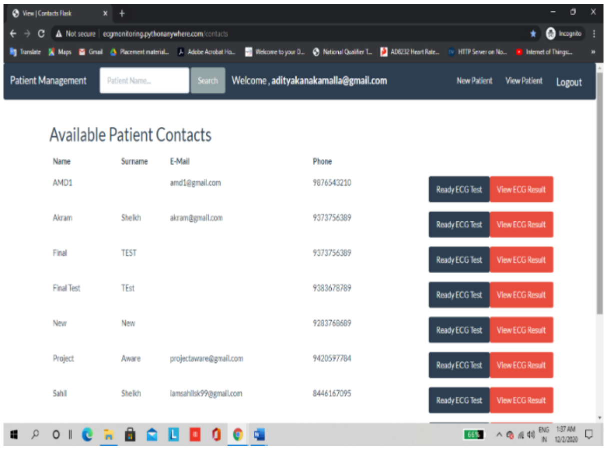


```
□ʕ□(ʕ)ʕʕʕjŐʕ;M_ʕ>ʕʕx□n□ʕz   > tʕPʕUyʕ`G}□H:WU□ʕ]ʕ□ʕʕgʕWIʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕʕ
*WM: [2] Connecting as wifi client...
*WM: [3] STA static IP:
*WM: [2] setSTAConfig static ip not set, skipping
*WM: [1] Connecting to SAVED AP: Sjeelani
*WM: [3] Using Password: 8790483265
*WM: [3] WiFi station enable
*WM: [3] enableSTA PERSISTENT ON
*WM: [1] connectTimeout not set, ESP waitForConnectResult...
*WM: [2] Connection result: WL_CONNECTED
*WM: [3] lastconxresult: WL_CONNECTED
*WM: [1] AutoConnect: SUCCESS
*WM: [1] STA IP Address: 192.168.0.102

connected... :)

>>>>>>>>>>>>>>>>>  Heart ECG Monitioring & Healthcare Syatem <<<<<<<<<<<<<<<<<<
******* Make sure you have register your details in web application ********
*** Make sure your device is connected with wifi and body with ECG module **
*WM: [3] unloading
```

Fig. 8 Serial monitor data of uploading data to cloud



Welcome , adityakanakamalla@gmail.com

Available Patient Contacts

Name	Surname	E-Mail	Phone		
AMD1		amd1@gmail.com	9876543210	Ready ECG Test	View ECG Result
Akram	Shelkh	akram@gmail.com	9373756389	Ready ECG Test	View ECG Result
Final	TEST		9373756389	Ready ECG Test	View ECG Result
Final Test	TEst		9383678789	Ready ECG Test	View ECG Result
New	New		9283768689	Ready ECG Test	View ECG Result
Project	Aware	projectaware@gmail.com	9420597784	Ready ECG Test	View ECG Result
Sahil	Shelkh	lamsahilsk99@gmail.com	8446167095	Ready ECG Test	View ECG Result

Fig. 9 Web main page.

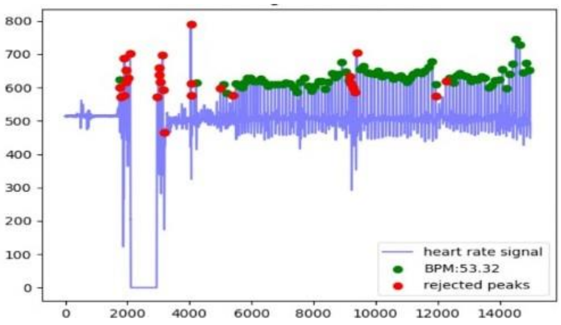
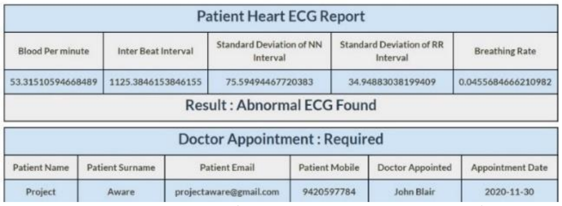


Fig. 10 ECG graph in webpage.



Patient Heart ECG Report				
Blood Per minute	Inter Beat Interval	Standard Deviation of NN Interval	Standard Deviation of RR Interval	Breathing Rate
53.31510594668489	1125.3846153846155	75.59494467720383	34.94883038199409	0.0455684666210982
Result : Abnormal ECG Found				

Doctor Appointment : Required					
Patient Name	Patient Surname	Patient Email	Patient Mobile	Doctor Appointed	Appointment Date
Project	Aware	projectaware@gmail.com	9420597784	John Blair	2020-11-30

Fig. 11 Detailed data analysis of ECG data and generation of appointment

The whole network is framed in a system that connects the patient with the clinic and helps in getting appointment easily. Figure 9 shows the web page (main) developed. The waveform generated by the proposed device can also be observed as shown in figure 10. After observing the patient's report, appointment will be generated based on the seriousness of the patient's condition, as shown in figure 11. This proposed system provides a platform that supports any elderly heart patient to run regular check-ups in home without travelling/going to a clinic. It is mostly applicable and beneficial for the patients in rural areas and hilly regions.

6- Conclusions

This paper intends to plan an ECG medical care framework dependent on IoT for patients who are experiencing issues in heart action. The framework uses the IoT application which is a simpler to book arrangement when it is irregular and furthermore without human presence, the doctors to screen the heart condition of their patients. The fundamental strength purposes of our proposed framework are anything but difficult to associate and use by anybody, can be conveyed anyplace, distantly utilized, savvy, and effectively prepared. This framework can be created in future by expanding the quantity of the anode for ECG sensor to get more precise outcomes, additionally can be stretched out by adding more sensors for estimating different boundaries, for example, pulse and pulses. Making a cloud information base will be exceptionally powerful to store the patient's wellbeing data including patient's wellbeing history, nonetheless, in each time the specialist plays out another ECG perusing the information will be straightforwardly put away in the patient's record in the data set, at that point it could be dissected, and the specialist can send a warning report to the patient, when they came to emergency clinic.

Acknowledgments

Acknowledgement: We thank the Department of Electronics and Communication Engineering (DST-funded), KL University for providing the facilities in Embedded and IoT lab and complete this project.

Conflict of Interest

This paper is not submitted or published in any journal and there is no conflict of interest.

References

[1] U. Satija, B. Ramkumar and M. Sabarimalai Manikandan, "Real-Time Signal Quality aware ECG Telemetry System for IoT-Based Health Care Monitoring," in IEEE Internet of Things Journal, vol. 4, no. 3, pp. 815-823, June 2017.

[2] R. K. Pathinarupothi, P. Durga and E. S. Rangan, "IoT-Based Smart Edge for Global Health: Remote Monitoring With Severity Detection and Alerts Transmission," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2449-2462, April 2019.

[3] M. A. Quiroz-Juárez, O. Jiménez-Ramírez, R. Vázquez-Medina, E. Ryzhii, M. Ryzhii and J. L. Aragón, "Cardiac Conduction Model for Generating 12 Lead ECG Signals With Realistic Heart Rate Dynamics," in IEEE Transactions on Nano Bioscience, vol. 17, no. 4, pp. 525-532, Oct. 2018.

[4] Yang Lei and Zhang Chao, "Design and Realization of Portable Rapid Electrocardiograph," China Medical devices, vol. 8, pp.II-13, 20 10.

[5] Yu Xuefei, "Theory and Design of Modern medical instrumentation," South China University of technology Press, vol. I, 2008.

[6] G. Xu, "IoT-Assisted ECG Monitoring Framework With Secure Data Transmission for Health Care Applications," in IEEE Access, vol. 8, pp. 74586-74594, 2020, doi: 10.1109/ACCESS.2020.2988059.

[7] B. M. Lee and J. Ouyang, "Intelligent Healthcare Service by using Collaborations between IoT Personal Health Devices," International Journal of Bioscience and Biotechnology, vol. 6, no. 1, p. 10, 2014.

[8] A. G. Ismaeel and E. K. Jabar, "Effective System for Pregnant Women using Mobile GIS," I J C A (0975 – 8887), vol. 64, no. 11, p. 7, 2013 2013.

[9] B. Padmavathi and S. T. Rana, "Implementation of IOT Based Health Care Solution Based on Cloud Computing," International Journal of Engineering and Computer Science, vol. 5, no. 9, p. 7, 2016.

[10] A. Ahamed, K. Hasan, and S. Alam, "Design and Implementation of Low-Cost ECG Monitoring System for the Patient Using Smartphone," presented at the (ICEEE), Rajshahi, Bangladesh, 2015.

[11] A. Škraba; A. Koložvari; D. Kofjač; R. Stojanović, Prototype of speech-controlled cloud-based wheelchair platform for disabled persons. Embedded Computing (MECO) 3rd Mediterranean Conference on, Budva, Montenegro. 15-19 June 2014, pp. 162 – 165.

[12] Karagoez, Mehmet Fatih; Turgut, Cevahir, "Design and Implementation of RESTful Wireless Sensor Network Gateways Using Node.js Framework," in European Wireless 2014; 20th European Wireless Conference; Proceedings of, vol., no., pp.1-6, 14-16 May 2014

[13] Carlos, R.; Coyle, S.; Corcoran, B.; Diamond, D.; Tomas, W.; Aaron, M.; Stroiescu, F.; Daly, K., "Web-based sensor streaming wearable for respiratory monitoring applications," in Sensors, 2011 IEEE, vol., no., pp.901-903, 28-31 Oct. 2011 doi: 10.1109/ICSENS.2011.6127168

[14] P.Kalaivani, T.Thamaraiselvi, and G. V. P.Sindhuja, "Real Time ECG and Saline Level Monitoring System Using Arduino UNO Processor," A J A S T, vol. 1, no. 2, p. 5, 2017.

[15] D. Barik and A. Thorat, "Issues of unequal access to public health in India," Frontiers in public health, vol. 3, p. 245, 2015.

[16] U. Lehmann, M. Dieleman, and T. Martineau, "Staffing remote rural areas in middle- and low-income countries: A literature review of attraction and retention," BMC Health Services Research, vol. 8, no. 1, p. 19, Jan 2008.

[16] A. Nordrum, "Italy launches a new wireless network for the internet of things," Dec 2017.

[17] B. Gayathri, K. Sruthi, and K. A. U. Menon, "Non-invasive blood glucose monitoring using near infrared spectroscopy," in 2017 International Conference on Communication and Signal Processing (ICCSP), April 2017, pp. 1139–1142.

[18] E. Nemati, M. J. Deen, and T. Mondal, "A wireless wearable ECG sensor for long-term applications," IEEE Communications Magazine, vol. 50, no. 1, pp. 36–43, January 2012

[19] S.Lavanya, G.Lavanya, and J.Divyabharathi, "Remote Prescription and I-Home Healthcare Based on IoT," presented at the I C I G E H T '17, Coimbatore, India, 02 November 2017, 2017.

[20] C. Lastre-Dominguez, Y. S. Shmaliy, O. Ibarra-Manzano, and M. Vazquez-Olguin, ‘‘Denoising and features extraction of ECG signals in state space using unbiased FIR smoothing,’’ IEEE Access, vol. 7, pp. 152166–152178, 2019.

[21] J. Zhang, A. Liu, M. Gao, X. Chen, X. Zhang, and X. Chen, ‘‘ECG-based multi-class arrhythmia detection using spatio-temporal attention-based convolutional recurrent neural network,’’ Artif. Intell. Med., vol. 106, Jun. 2020, Art. no. 101856.

[22] S. K. Pandey, R. R. Janghel, and V. Vani, ‘‘Patient specific machine learning models for ECG signal classification,’’ Procedia Comput. Sci., vol. 167, pp. 2181–2190, Jan. 2020.

[23] S. Nurmaini, A. Darmawahyuni, A. N. Sakti Mukti, M. N. Rachmatullah, F. Firdaus, and B. Tutuko, ‘‘Deep learning-based stacked denoising and autoencoder for ECG heartbeat classification,’’ Electronics, vol. 9, no. 1, p. 135, Jan. 2020.

[24] M. Faezipour and M. Faezipour, ‘‘System dynamics modeling for smartphone-based healthcare tools: Case study on ECG monitoring,’’ IEEE Syst. J., early access, Jul. 23, 2020, doi: 10.1109/JSYST.2020. 3009187.

[25] H.-T. Chiang, Y.-Y. Hsieh, S.-W. Fu, K.-H. Hung, Y. Tsao, and S.-Y. Chien, ‘‘Noise reduction in ECG signals using fully convolutional denoising autoencoders,’’ IEEE Access, vol. 7, pp. 60806–60813, 2019.

[26] F. Liu, C. Liu, L. Zhao, X. Zhang, X. Wu, X. Xu, Y. Liu, C. Ma, S. Wei, Z. He, J. Li, and E. N. Yin Kwee, ‘‘An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection,’’ J. Med. Imag. Health Informat., vol. 8, no. 7, pp. 1368–1373, Sep. 2018.

[27] A. E. Curtin, K. V. Burns, A. J. Bank, and T. I. Netoff, ‘‘QRS complex detection and measurement algorithms for multichannel ECGs in cardiac resynchronization therapy patients,’’ IEEE J. Transl. Eng. Health Med., vol. 6, pp. 1–11, 2018.

[28] S. M. Mathews, C. Kambhamettu, and K. E. Barner, ‘‘A novel application of deep learning for single-lead ECG

classification,'' Comput. Biol. Med., vol. 99, pp. 53–62, Aug. 2018.

[29] C. Venkatesan, P. Karthigaikumar, A. Paul, S. Satheeskumaran, and R. Kumar, ''ECG signal preprocessing and SVM classifier-based abnormality detection in remote healthcare applications,'' IEEE Access, vol. 6, pp. 9767–9773, 2018.

[30] J. M. Bote, J. Recas, F. Rincon, D. Atienza, and R. Hermida, ''A modular low-complexity ECG delineation algorithm for real-time embedded systems,'' IEEE J. Biomed. Health Informat., vol. 22, no. 2, pp. 429–441, Mar. 2018.

[31] X. Tang, Q. Hu, and W. Tang, ''A real-time QRS detection system with PR/RT interval and ST segment measurements for wearable ECG sensors using parallel delta modulators,'' IEEE Trans. Biomed. Circuits Syst., vol. 12, no. 4, pp. 751–761, Aug. 2018.

[32] V. H. Goh and Y. Wen Hau, ''Android-based mobile application for homebased electrocardiogram monitoring device with Google technology and Bluetooth wireless communication,'' in Proc. IEEE-EMBS Conf. Biomed. Eng. Sci. (IECBES), Dec. 2018, pp. 205–210.

[33] U. Satija, B. Ramkumar, and M. S. Manikandan, ''A review of signal processing techniques for electrocardiogram signal quality assessment,'' IEEE Rev. Biomed. Eng., vol. 11, pp. 36–52, 2018.

[34] U. Iqbal, T. Ying Wah, M. Habib Ur Rehman, and Q.-U.-A. Mastoi,''Usage of model driven environment for the classification of ECG features: A systematic review,'' IEEE Access, vol. 6, pp. 23120–23136, 2018.

Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 20-27

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Secure Key Management Scheme for Hierarchical Network Using Combinatorial Design

Siddiq Iqbal[1*], Sujatha B R[2]

[1].Department of Electronics & Telecommunication B.M.S. Institute of Technology & Management, Bengaluru, India
[2].Department of Electronics & Communication Malnad College of Engineering, Hassan, India

Received: 31 Jan 2021 / Revised: 09 Aug 2021/ Accepted: 09 Sep 2021
DOI: https://doi.org/10.52547/jist.15691.10.37.20

Abstract

The wireless sensor network (WSN) signifies to a gathering of spatially spread and committed sensors for observing and logging the physical states of the environment and for organizing the information gathered at the central Base station. Many security threats may affect the functioning of these networks. Security of the data in the system depends on the cryptographic procedure and the methods where encryption and decryption keys are developed among the sensors. Symmetric key foundation is one of the best applicable ideal models for safe exchanges in WSNs. The main goal is to improve and evaluate certain issues, such as node attack, to provide better key strength, connectivity, security for node interaction, and throughput. Uniform Balanced Incomplete Block Design (UBIBD) is used to generate the keys allocated by the base station to the cluster head. The cluster head distributes keys to its members using Symmetric Balanced Incomplete Block Design (SBIBD), and the keys are refreshed on a regular basis to avoid out-of-date entries. In wireless sensor networks, compromised nodes can be used to inject false reports. The concept of interacting between sensor nodes using keys and establishing a secure connection aids in ensuring the network's security.

Keywords: Wireless Sensor Networks; Combinatorial Design; Key Management; Key Distribution; Key Refreshment; Balanced Incomplete Block Design.

1- Introduction

A wireless sensor network has a massive number of minute nodes, each with its own set of energy limitations and computing difficulties. WSN offers a wide range of applications, including military, medical, and residential appliance management. In general, sensor devices have resource constraints; these constraints govern the design of WSNs and, as a result, influence the security level of sensor networks. Transmission distance, restricted processing power, battery lifetime, limited memory, random scattering of nodes, and bandwidth are examples of such constraints [3].

- Transmission distance: In sensor networks, sensors have transmission range which is restricted. Longer the distance of transmission more energy will be consumed. Thus, the communication range for sensor nodes is controlled due to the limited available resources.
- Limited processing power: A sensor node in a network has its own special hardware and software architecture. During the manufacturing process of a sensor node, the main aim is to save the energy consumed by the node, to achieve this the hardware and software architecture should be as simple as possible.
- Battery life: As the sensor devices are operating usually in outdoor environment, they are completely dependent on battery power. Sensor nodes rely solely on their battery for power, so once it runs out, the status of the sensor node is considered inactive.
- Limited memory: Sensor node just like any other device has its own unique operating system. This operating system is preloaded into the memory of the sensor and occupies a part of storage. The remaining memory of the node should be managed efficiently. For efficient utilisation of memory the storage space occupied by the cryptographic keys which are used in the network for secure transmission of data should be low. Less space occupied by the keys gives space for other processing operations to be carried out without any limitation.
- Prior deployment knowledge: In most of the applications, the deployment of sensor nodes is in high risk areas, such as war zones. It is critical to select the deployment method, as usually the deployment is random or carried out in real-time. This problem adds to location information of a sensor node being insufficient or most likely unavailable. During the start of key assignment

✉ **Siddiq Iqbal**
Siddiq@bmsit.in

process, the key management procedures should be independent of the sensor location.

- Bandwidth: In WSN, the key establishment process is constrained by sensor node constraints. As previously stated, the hardware architecture of sensor nodes is straightforward. Because such nodes' transmitters have limited capacities, they can't send a large amount of data all at once. As a result, the bandwidth capacity of sensor nodes is limited. As a result, key management techniques should consider this issue.

Because of these limits and the vulnerability of the WSN to attackers, protecting the sensor network is seen as a big task. To achieve this task, many algorithms have been suggested in the literature. WSN necessitates the use of a cryptography algorithm, which must be appropriately chosen, and the most crucial component for those algorithms is to resolve the Key agreement or management problem, which would be considered necessary provide the encrypted and authenticated data transmission among sensor nodes in order to have a channel with good security. Many security needs, such as confidentiality, integrity, authenticity, and availability, must be met by the cryptography algorithm (CIAA). To counteract attacks in WSN, a key management scheme is used [1]. Key distribution can be accomplished in a variety of ways, the simplest of which is to allot a single unique secret key to all the nodes in the network. However, if an adversary manages to obtain this secret key, the entire network's security is jeopardised. A more realistic way to key pre-distribution is to assign unique pair-wise keys per each link amongst sensor nodes in the network. The resiliency of such an approach is quite high, because compromising any of pair-wise key has no impact on the remaining network. However, because each node has to maintain keys with all of the other sensor nodes in the network, the pair-wise keys-based approach increases key storage overhead. Combinatorial design-based key pre-distribution is a kind of compromise, where we sacrifice network robustness in exchange for lower storage overhead. This architecture includes allocating a pool of keys to each of the sensor nodes, with each pair of key sets sharing some keys.

The remaining paper is structured as follows: Sect. 2 will go over some of the previous work in the field of key management. In Sect. 3, we will state the proposed scheme, which includes different phases of key management. In Sect. 4, we will explain the performance evaluation. In Sect. 5, we thoroughly explain the simulation results. In Sect. 6, Finally, we will present the conclusion of our paper.

2- Related Works

The process of keeping track of different cryptographic keys in a cryptosystem is known as key management. Key generation, exchange, storage, handling, crypto-shredding, and replacement are all aspects of key management. The generation and distribution of keys is carried out using BIBD [2] [4]. The essence of the key management scheme's major updation based on a unified design is illustrated, two main update techniques are proposed, and the results of the three approaches are analysed in two ways. Finally, the two strategies are extended to other combinatorial plan-based key management systems, and the second strategy is enhanced. Key management approaches based on combinatorial plans receive a lot of attention [16]. Key administration schemes that do not include key updates will be less protected in the long run [7]. Despite the fact that the NBIBD was first published in 1967 and sparked widespread interest, they were virtually not known in combinatorial writing outside of the competitive literature, and their entire combinatorial inferences were not sufficient. The reviews and extensions of numerical understanding on NBIBDs were discussed. For NBIBDs, isomorphism and auto orphisms are specified, as well as building methods. NBIBD is divided into several categories, each of which is defined and illustrated [8]. State-of-the-art cryptography procedures are used to secure computer networks. While it has been claimed that designing good cryptographic procedures is the simplest approach to secure a large scale network, it seems that security concerns in algorithms and their executions are frequently observed [10] [11]. The development of wireless data transmission in the initial twentieth century is credited with the first cryptographic explosion; An adversary may clearly read radio transmission just as easy as a true recipient. [9].

Block constructions are especially useful in related experiments when working with heterogeneous test rings to increase the impact of treatment evaluations. The Embedded Balanced Incomplete Block Design (NBIBD) is a concept with blocks of two systems, one of which is the original location and the other of which is the ultimate target blocks that neglect one of the two systems and leaving a balanced incomplete block design. For gathering a number of characteristics, a novel methodology for generating the structure of built-in balanced incomplete blocks is devised [14].

The nested block design is characterised as a floor plan with two systems of blocks, the second of which is positioned at the base (two squares of the second system are present in each square of the first), with the final goal being a balanced incomplete block on both frames. an image in which the squares correspond to the squares of another system. The list of these plans is only repeated fifteen times in each treatment. Yates' extension to retrieve block data in balanced incomplete block diagrams is the subject of this investigation [15]. The analysis is made for a balanced incomplete scheme that is typically developed. Sec-LEACH [17] combines TESLA and random key pre-distribution to produce a secure communication system that is efficient and resists few attacks. Sec-LEACH is being used to compare with the proposed method.

3- Proposed Scheme

The suggested key management is divided into phases (shown in Figure.1).

- Node registration.
- Key generation and Key distribution.
- Key refreshment.

3-1- Node Registration

Initially, the base station registers all the nodes which have been deployed, before key distribution [6]. During this phase, the base station passes different registration messages and assign IDs to all the nodes in the network. Every node in the cluster transmits the identification parameters to all the other members in the cluster. A node in a cluster has three association parameters, the node ID (distinct for every node), the cluster ID (distinct for each cluster of sensor nodes), location of the node.

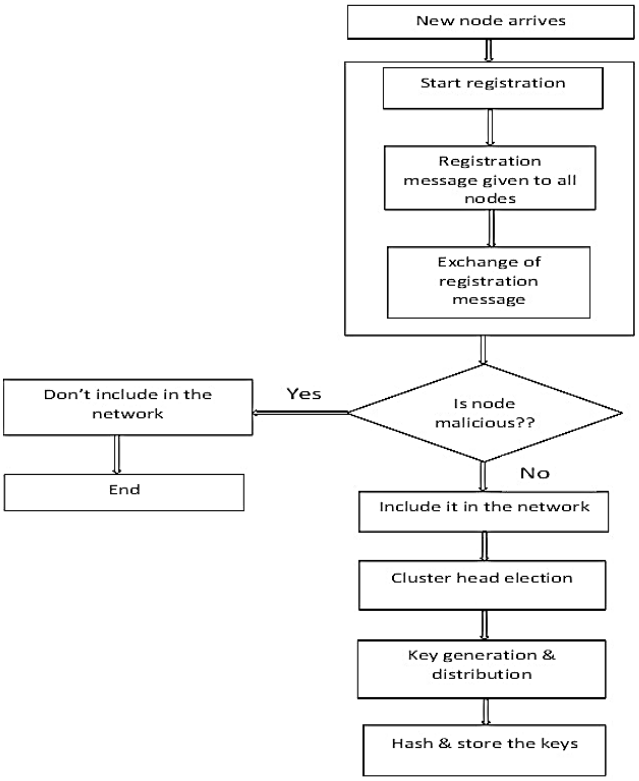

New node arrives
Start registration
Registration message given to all nodes
Exchange of registration message
Is node malicious??
Yes
No
Don't include in the network
End
Include it in the network
Cluster head election
Key generation & distribution
Hash & store the keys

Fig. 1 Proposed scheme

The base station transmits information of every node to all the cluster members along with the registration messages transmitted to the nodes. The Neighbor Information Table holds the information of cluster members [5]. A node in the cluster first sends a message to its closest neighbors, followed by sending the message to all other members in the cluster. Every sensor node sends the message in addition to its ID. When a node is passing the message further, it sends its own ID along with the forwarded message. All the cluster nodes receive the registration messages from each cluster member.

A node in a cluster has one more table which stores registration messages and IDs of cluster members. When there is a mismatch in the information of registration message and ID received from any node, the node forwards the location of such a node to the base station. The malicious nature of a node is confirmed by the base station after analysing the data transmission and reception pattern of the suspicious node and if found to be malicious then the location is added to the blacklist. All other nodes which are tested not to be suspicious are registered with the base station. The nodes that have registered become a part of the network's subsequent processes.

Nodes are detected as malicious and removed based on their location information. Depending on its location, the node closest to the cluster's centre announces itself as the cluster head and sends a declaration message to the base station as well as all of the cluster's members. The acknowledgement received from the base station and cluster members is used to identify a node as the cluster head. When a node is marked as CH, a link is made between the BS, CH, and members nodes.

3-2- Key generation and distribution

Cluster head keys are created using the UBIBD (9,3,1) design by the base station, while member node IDs are generated using the SBIBD (45,12,3) design, where (9,3,1) and (45,12,3) are v (number of distinct values), b (number of blocks), and lambda (number of blocks in which two distinct values appear) respectively.

Following steps are used for key generation:

- Size of CH keys is 128 bit, which are generated randomly.
- Member node IDs are 128 bit random number.
- In order to generate authentication key and MN keys, the CH keys are used (Figure 2).

1. Consider the following three keys that belong to a CH based on the UBIBD (9, 3,1) architecture (each block of the design contains three distinct keys, which are given to each cluster head).
2. Each block in the UBIBD design has its own set of keys, pairs chosen at random are all unique and utilised to generate unique login keys for the member nodes of each cluster.
3. Assume [a, b, c] are cluster head keys, Let [c, a] be a pair of 256-bit authentication keys created at random by the product of c and a.

   i.e., $AK = c * a$

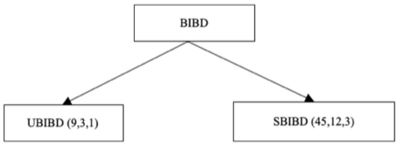


Fig. 2 Key Generation

The 256 bit AK and the data messages are used to generate the hashed output. Using HMAC SHA-256, this hashed data will be used for the authentication and integrity of the data.

4. MN keys are generated using one of the three CH keys, as well as the other two keys.
5. In the set [a, b, c] CH keys, the key which is not used to calculate AK is ‘b’ and the keys used in AK are c & a.
6. ‘b’ gets multiplied with both ‘c’ and ‘a’ separately to generate sub products and then the resulting products are multiplied to obtain a disntict multiple. i.e.,

   $prod\_1 = b * c$
   $prod\_2 = b * a$
   $Umultiple=prod\_1 * prod\_2$

   The secret key (SK) is extracted from the first half of unique multiple (256 bit MSB).

7. To generate MN keys, multiply the resultant multiple by the 128-bit MN IDs.

   MN keys = SK * ID (SBIBD)

As a result of all of the preceding calculations, the final result is:

- CH keys - 128 bit
- Authentication Key - 256 bit
- MN keys - 382 bit

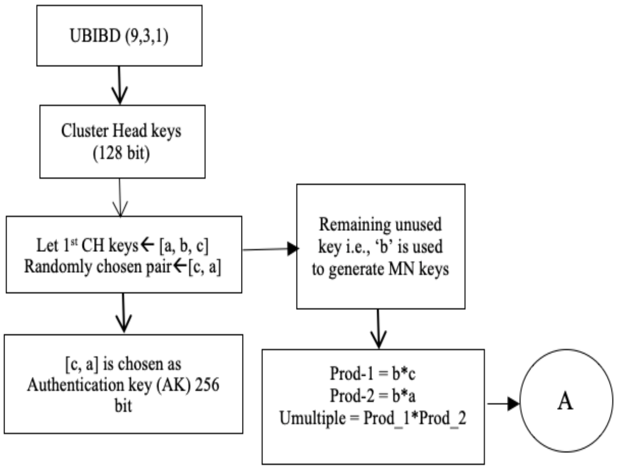

UBIBD (9,3,1)
Cluster Head keys (128 bit)
Let 1st CH keys← [a, b, c] Randomly chosen pair←[c, a]
Remaining unused key i.e., ‘b’ is used to generate MN keys
[c, a] is chosen as Authentication key (AK) 256 bit
Prod-1 = b*c
Prod-2 = b*a
Umultiple = Prod_1*Prod_2
A

Fig. 2a Key Generation, UBIBD (9,3,1)

3-3- Key refreshment

XOR computations are used to refresh keys (Fig 3).

- • A total of thirteen 128-bit random numbers are generated. (Magic words: one WCH and twelve WMN).
  - W CH- used to refresh cluster head keys.
  - W MN - used for member node’s key refreshment

Key refreshment for CH:

- Refreshment of CH key is carried by using one of the thirteen magic words (WCH). WCH is circularly shifted four times, and the resulting bit is R.

  R = circular shift (WCH , 4)
- XOR operation is carried out on the old key with the magic word WCH and the result obtained is defined as Kn.

  $Kn = Old\_Key\ CH \oplus WCH$
- A new refreshed key is generated by applying XOR on ‘R’ with the previous result Kn.

  $K\ CH\_refreshed = R \oplus Kn$

MN key refreshment:

- From the thirteen words, twelve magic words are used for key refreshment of member node.
- The XOR operation is conducted using corresponding W-MN in each cluster, and IDs are renewed. i.e.,

  R = circular shift (W-MN , 4)
  $Kn = Old\_key\ MN \oplus W\text{-}MN$
  $MN\ refreshed\ ID = R \oplus Kn$
- MN refreshed key is generated by multiplying the Secret key (SK) generated by the new CH key with its corresponding MN refreshed ID.
- MN refreshed key = SK * MN refreshed ID

As a result, all Cluster head and member node keys are refreshed on a regular basis, ensuring that the UBIB and SBIB designs are met.

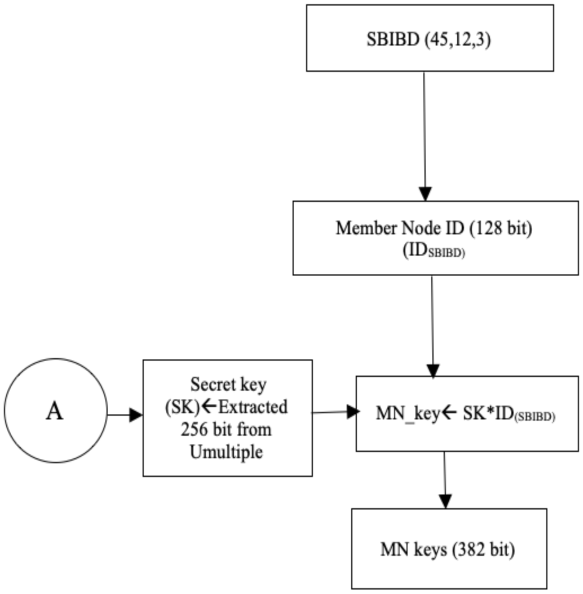

SBIBD (45,12,3)
Member Node ID (128 bit) (ID_{SBIBD)}
A
Secret key (SK)←Extracted 256 bit from Umultiple
MN_key← SK*ID(SBIBD)
MN keys (382 bit)

Fig. 2b Key Generation, SBIBD (45,12,3)

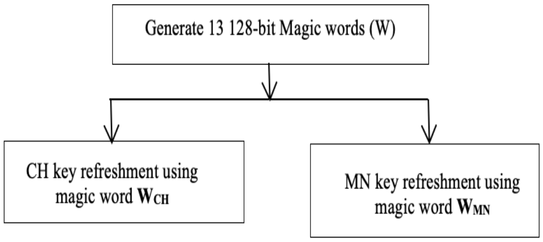

Generate 13 128-bit Magic words (W)
CH key refreshment using magic word WCH
MN key refreshment using magic word WMN

Fig. 3 Key Refreshment

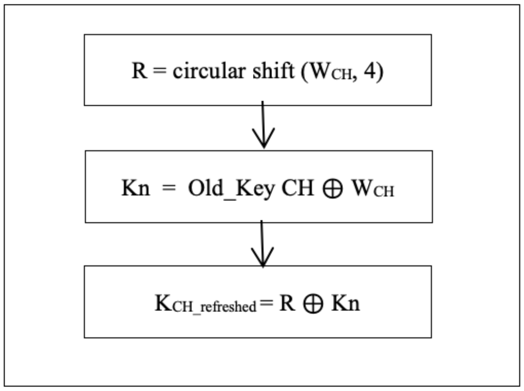


Fig. 3a CH key refreshment using magic word W_{CH}

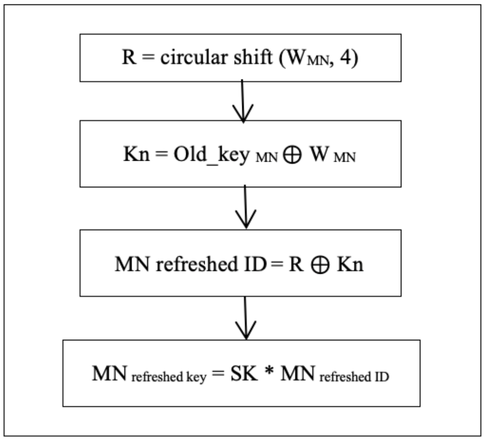

R = circular shift (WMN, 4)
Kn = Old_key MN ⊕ W MN
MN refreshed ID = R ⊕ Kn
MN refreshed key = SK * MN refreshed ID

Fig. 3b MN key refreshment using magic word W_{MN}

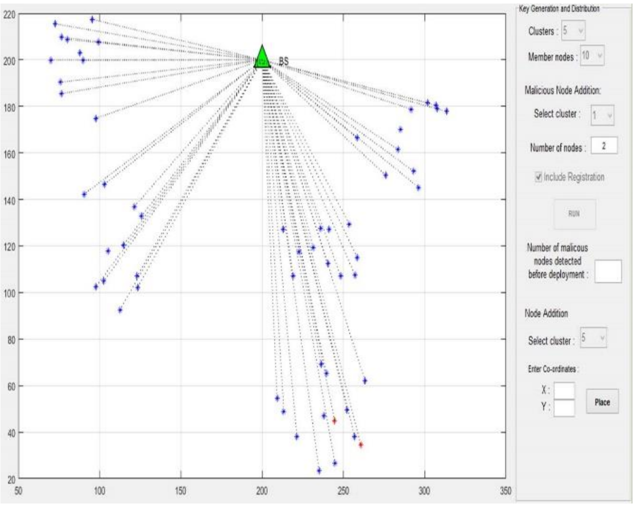
4- Performance Evaluation

The performance of the proposed scheme presented in this paper is evaluated in this section. For implementation and performance analysis, we used the MATLAB tool for programming. A network with a size of 200m x 200m and a transmission range of 30m has been considered. The member nodes are located within the range of all the clusters, with the base station in the middle. For simulation, 500 rounds are taken and the nodes with initial energy of 0.5J is considered. Total number of nodes considered for simulation is 50 and two nodes are assumed to be compromised. Whichever node that is not registered is compromised. Randomly few nodes are considered to be compromised. The graphs show the average results. The modeling results are compared.

The metrics used for performance analysis are:

- Throughput - The total data sent over the network.
- Security – Security is given to the network by eliminating malicious nodes [5].
- Energy – Energy remaining in the network at the end of each round.

5- Simulation Results



Key Generation and Distribution

Clusters : 5

Member nodes : 10

Malicious Node Addition:

Select cluster : 1

Number of nodes : 2

☑ Include Registration

RUN

Number of malicous nodes detected before deployment :

Node Addition

Select cluster : 5

Enter Co-ordinates :

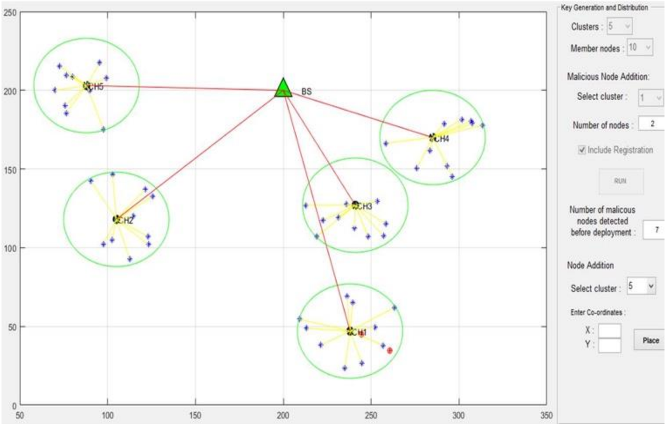
X :

Y :

Place

Fig. 4 Registration of nodes

Figure 4 depicts the registration process, which identifies and eliminates malicious nodes. As shown in the diagram, malicious items are highlighted in red and are not registered. Unregistered nodes will not be included in network since the base station issues keys to all registered nodes.



Key Generation and Distribution

Clusters : 5

Member nodes : 10

Malicious Node Addition:

Select cluster : 1

Number of nodes : 2

☑ Include Registration

RUN

Number of malicous nodes detected before deployment : 7

Node Addition

Select cluster : 5

Enter Co-ordinates :

X :

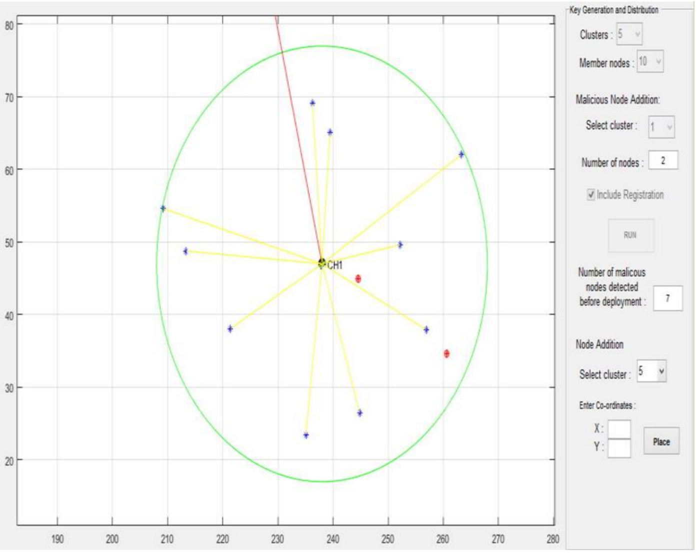
Y :

Place

Fig 5. GUI Showing Base Station and Clusters

Figure 5 depicts the complete deployment of the network, including the base station, cluster heads, and member nodes. A malicious node can be added to any of the clusters by the user. The GUI displays a drop box where we can specify the number of member nodes, clusters, and suspicious nodes. The box which shows the number of malicious nodes which have been identified, here we can also insert new nodes.

Fig.6 shows the closer view of the cluster 1. The cluster consists of 10-member nodes and two malicious nodes which are red in color. This image shows the cluster after the identification and elimination of malicious nodes.



Key Generation and Distribution

Clusters : 5

Member nodes : 10

Malicious Node Addition:

Select cluster : 1

Number of nodes : 2

☑ Include Registration

RUN

Number of malicous nodes detected before deployment : 7

Node Addition

Select cluster : 5

Enter Co-ordinates :

X :

Y :

Place

Fig. 6. Closer view of the cluster

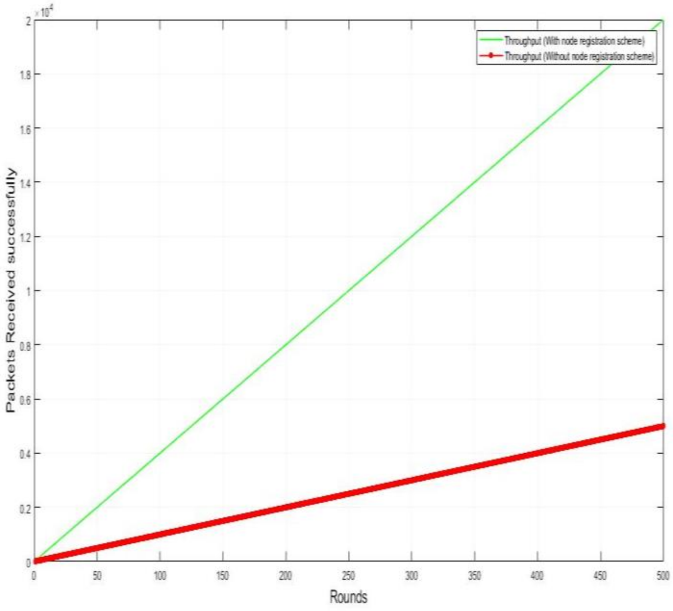


Fig 7. Throughput graph

The fig. 7 shows the throughput in the network for 500 rounds. In the above figure the red line represents the packets received successfully without the registration scheme that is 5000 for all the rounds, green line which is increasing linearly represents the packets received successfully with registration scheme which is 20000. Without employing node registration process the throughput of the network will be reduced as a result of the presence of malicious nodes.

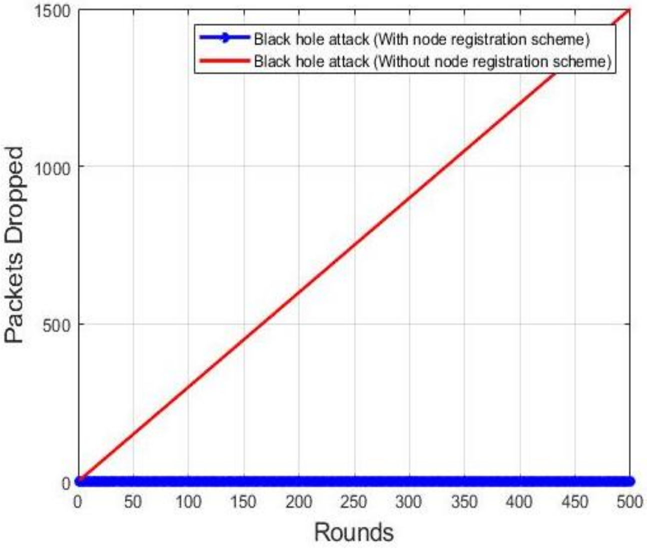


Fig 8. Packets dropped graph

The fig.8 shows the packets dropped in the network for 500 rounds. The red line in the graph is increasing linearly shows that 1500 packets are dropped without the registration scheme and the blue line in the graph shows the packets dropped with registration scheme is 0. Without including the node registration process, the number of packets dropped in every round of communication will be more than that when only registered nodes are involved in communication.

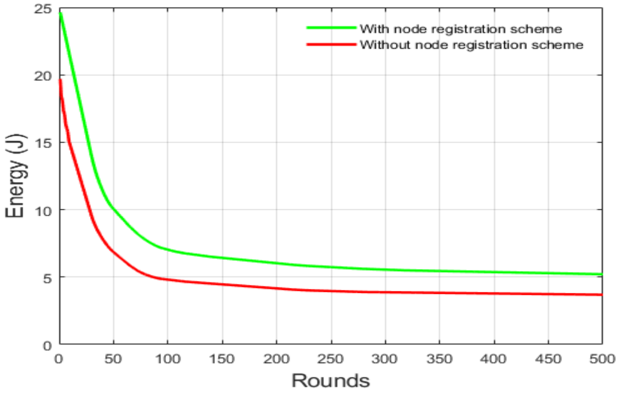


Fig 9. Energy graph

The fig.9 shows the energy graph of the network, which shows the amount of energy consumed as the number of rounds increase. Since there are 50 nodes, each node as 0.5 J of energy, so total energy is 25 J. From the graph, the total energy of the network at the end of 500 rounds of communication involving only registered i.e. non-malicious nodes was found to be around 5J whereas, that when involved the malicious nodes was found to be around 4.8J. The presence of malicious node negatively influences the overall energy consumption of the network.

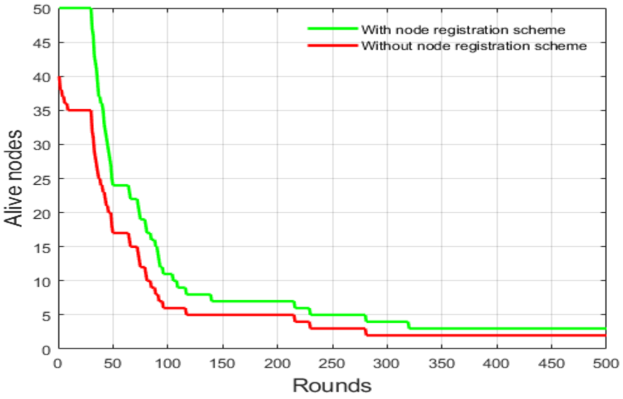


Figure 10. Alive nodes graph

The fig.10 shows the number the nodes alive in the network. Green line indicates there are 50 valid nodes, red line indicated there are 10 nodes out of 50 which are not registered. It was observed that with node registration scheme, since the influence of malicious nodes will be minimized, the number of alive nodes after 500 rounds of communication was found to be more (2 alive nodes) than that without node registration scheme (1 alive node). It is desired to have the nodes alive for longer duration which in turn would extend the lifetime of the network.

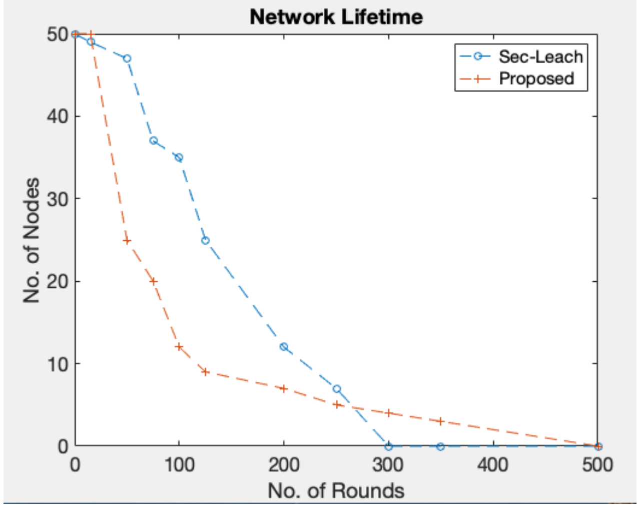


Figure 11. Network Lifetime Analysis

Figure 11 analyzes the results obtained in terms of network lifetime. Because there are various approaches to analysing the lifetime of a WSN, the number of alive nodes is taken into account here. From the figure, it is depicted that the presented method extends the network lifetime and extends the death of the nodes in WSN. Overall the network life time has been increased in the proposed method compared to existing method.

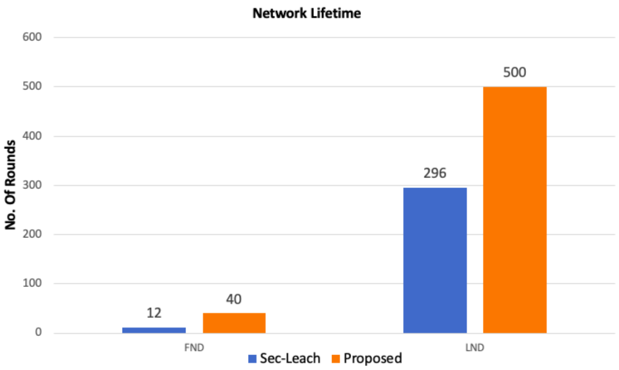


Figure 12. Network Lifetime

Another method for determining network lifetime analysis is to identify the first and last nodes to die (FND and LND) in a WSN. Figure 12 depicts the proposed and existing methods' network lifetime analysis in terms of FND and LND. Figure 12 shows that the presented technique, when compared to other methods, delayed the death of the first node . In Sec-LEACH FND occurs after 12 rounds, whereas the presented model's FND occurs after 40 rounds. Similarly, Sec-LEACH has an LND of 296 rounds, whereas the presented model has an LND of 500 rounds.

6- Conclusions

The proposed approach was implemented utilising static wireless nodes. When compared to existing schemes, this scheme provides better connectivity, coverage, key strength, and attack resistance. Because the primary benefit of sensor nodes is their mobility, they can be integrated into the system through the use of advanced routing protocols that boost system efficiency and support mobility. The impacts of increased node density must be investigated further. The proposed technique could be developed even more to provide higher resilience to a wider range of threats. Cluster heads are assumed to have high resistance against attacks, even if a cluster head is compromised the cells connected to that particular Cluster head are compromised. By making further advancements in this area, the system can be made more reliable.

References

[1] Alok Kumar, Neha Bansal, and Alwyn R. Pais. "New key pre-distribution scheme based on combinatorial design for wireless sensor networks." IET Communications 13, no. 7, 2019, pp. 892-897.

[2] Yang, Chin-Nung, Ting-Ju Lin, Song-Yu Wu, Shin-Shang Lin, and Wei Bi. "Cost Effective Hash Chain Based Key Pre-Distribution Scheme for Wireless Sensor Network." In 2018 IEEE 18th International Conference on Communication Technology (ICCT), pp. 518-522. IEEE, 2018.

[3] Khawla Naji Shnaikat and Ayman Ahmed Al Qudah, "Key management techniques in wireless sensor networks", International Journal of Network Security & Its Applications (IJNSA) Vol.6, No.6, November 2014, pp. 49-63.

[4] G. Choi and I. Lee, "A key distribution system for user authentication using pairing-based in a WSN," 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), 2017, pp. 1-4.

[5] Ruhul Amin, and G. P. Biswas. "Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment." Wireless Personal Communications 84, no. 1, 2015, pp. 439-462.

[6] M. M. M. Fouad, M. M. Mostafa and A. R. Dawood, "A Pairwise Key Pre-distribution Scheme Based on Prior Deployment Knowledge," 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks, 2011, pp. 184-189.

[7] Chonghuan Xu, and Weinan Liu, "Key Updating Methods for Combinatorial Design Based Key Management Schemes", Hindawi Publishing Corporation, Journal of Sensors, 2014.

[8] J.P. Morgana, D.A. Preeceb and D.H. Reesb, "Nested balanced incomplete block designs", Discrete Mathematics, vol. 231, 2001.

[9] Bart Preneel, "Cryptography for network security: failures, successes and challenges." In International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, pp. 36-54. Springer, Berlin, Heidelberg, 2010.

[10] Mukund R. Joshi, and Renuka Avinash Karkade, "Network security with cryptography." International Journal of Computer Science and Mobile Computing 4, no. 1, 2015, pp. 201-204.

[11] P. M. Wightman and M. A. Labrador, "A3: A Topology Construction Algorithm for Wireless Sensor Networks," IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference, 2008, pp. 1-6.

[12] Shalini M S, Hemanth S R, "EKM-CI: Effectual key administration in dynamic wireless sensor network", International Research Journal of Engineering and Technology (IRJET), May 2018, pp.226-228.

[13] V. S. Janani and M. S. K. Manikandan, "Enhanced Security Using Cluster Based Certificate Management and ECC-CRT Key Agreement Schemes in Mobile Ad hoc Networks." Wireless Personal Communications 97, no. 4, 2017, pp. 6131-6150.

[14] A. Dey, U. S. Das, and A. K. Banerjee, "Construction of nested balanced incomplete block designs." Calcutta Statistical Association Bulletin 35, no. 3-4, 1986,pp. 161-168.

[15] D. A. Preece, "Nested balanced incomplete block designs." Biometrika 54, no. 3-4, 1967, pp. 479-486.

[16] Kazeem A. Osuolale and Oluwaseun A. Otekunrin. "An algorithm for constructing symmetric ((r+ 1) v, kr, kλ) BIBDs from affine resolvable (v, b, r, k, λ) BIBDs." Annals. Computer Science Series 12, no. 2, 2014.

[17] Sariga Arjunan, Sujatha Pothula and Dhavachelvan Ponnurangam, "F5N- based unequal clustering protocol (F5NUCP) for wireless sensor networks." International Journal of Communication Systems 31, no. 17, 2018, e3811.

Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 28-36

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

An Agent Based Model for Developing Air Traffic Management Software

Mahdi Yousefzadeh Aghdam[1] ,Seyed Reza Kamel Tabbakh[2]*,Seyed Javad Seyed Mahdavi Chabok[2] ,Maryam Kairabadi[1]

[1].Department of Computer Engineering, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran
[2].Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Received: 23 Nov 2020/ Received: 04 Aug 2021/ Accepted: 03 Sep 2021
DOI: https://doi.org/10.52547/jist.15635.10.37.28

Abstract
The Air Traffic Management system is a complex issue that faces factors such as Aircraft Crash Prevention, air traffic controllers pressure, unpredictable weather conditions, flight emergency situations, airplane hijacking, and the need for autonomy on the fly. agent-based software engineering is a new aspect in software engineering that can provide autonomy. agent-based systems have some properties such: cooperation of agents with each other in order to meet their goals, autonomy in function, learning and Reliability that can be used for air traffic management systems. In this paper, we first study the agent-based software engineering and its methodologies, and then design a agent-based software model for air traffic management. The proposed model has five modules .this model is designed for aircraft ,air traffic control and navigations aids factors based on the Belief-Desire-Intention (BDI) architecture. The agent-based system was designed using the agent-tool under the multi-agent system engineering (MaSE) methodology, which was eventually developed by the agent-ATC toolkit. In this model, we consider agents for special occasions such as emergency flights' and hijacking airplanes in airport air traffic management areas which is why the accuracy of the work increased. It also made the flight's sequence arrangement in take-off and landing faster, which indicates a relative improvement in the parameters of the air traffic management

Keywords: Agent-Based Software Engineering; Agent-Based Modeling; BDI Architecture; Enterprise-Oriented Software Engineering; MaSE Methodology.

1- Introduction

Air traffic management system In addition to arranging and managing airspace controlled faces a complex distributed system with unpredictable cases that can disrupt the order of the ordinary stream. The decision to make optimal air traffic layout in such situations is matters to the routing independence and intelligent prevention of collision for each flight .agent-based software engineering provides facilities for designing and building complex distribution systems. In this methodology, the agent-oriented approach to designing distributed and complex air traffic management systems were studied. The system is faced with factors such as aircraft crash prevention, air traffic controllers pressure, unsuspended weather conditions, flight emergencies, airplane hijacking, and the need for autonomy on the fly. for this purpose, it is essential to develop new systems for air traffic flow [2]. The new approaches in the field of agent-based software engineering and Multi-agent Systems (MAS) have also brought us a new solution. These systems can simulate the air traffic control models based on the autonomy of operation and the relative autonomy of planes whether within the scope of airport control or in other areas in controlled space. In this paper, we first study agent-based software engineering and its methodologies, and then design an agent-based software model for air traffic management. The proposed model has five modules .this model is designed for aircraft, air traffic control and navigationsaids factors based on the belief- desire- Intention (BDI) architecture. The agent based system was designed using the Agent-Tool under the Multi-agent system engineering (MaSE) methodology, which was eventually developed by the agent- ATC toolkit. In this model, we consider agents for special occasions such as emergency flights' and hijacking airplanes in airport air traffic management areas which is why the accuracy of the work increased. It also made the flights' sequence arrangement in take-off and landing faster, which indicates a relative improvement in the parameters of the air traffic management.

✉ **Seyed Reza Kamel Tabbakh**
rezakamel@computer.org

The following sections of the paper have been structured, with Sect2. "Literature review" reviews the previous methods in the agent-based software of ATM field, Sect3. "BDI architecture" and sect4. "agent tool methodology design" describes the methodology of the agent system. In Sect5. "The MaSE- based analysis and design steps in agent tool" explain about how to Implementation of system. then in Sect6. "Analysis and design of the proposed model" implementation of the proposed model of air traffic management model and evaluates it; in addition, Sect. 8 has been dedicated to a conclusion.

2- Literature Review

The study of the behavior of air traffic management (ATM) systems using modeling systems and simulation tools can aid in the development and optimizing of new methods of improving ATM performance. The agent-based software system was applied as a solution for developing models that capture air traffic decisions and interactions with an adequate level of detail [29].agent-based models gained popularity as an effective approach to modeling and simulating specific dynamics of ATM systems. Different agent-based approaches were applied in ATM studies concerning air traffic control (ATC). These agent-based models represent air traffic processes such as vehicle trajectories, collision avoidance, airport operation, etc. In this category of models, we can mention Air MIDAS [30], ACES [31], AgentFly [32] and tools such as AirTop [33] and CAST [34]. These approaches are capable of simulating how air traffic controllers manage aircraft as they fly and move in the airports with a high level of detail. For example, Air MIDAS simulates the behavior of the final approach of aircraft in the terminal airspace and their interaction between the pilots and the flight controllers for risk evaluation. ATM studies related to air traffic flow management (ATFM) have recently been performed using also agent-based approaches. In general, they simulate ATFM on the day of the flight [35].

3- BDI Architecture

The BDI architecture or the belief-desire-intention (BDI) architecture is based on three concepts of belief, desire, and program. Belief is a statement of what the agent thinks about. BDI agents have a set of beliefs that are like the set of facts in law-based systems [4].

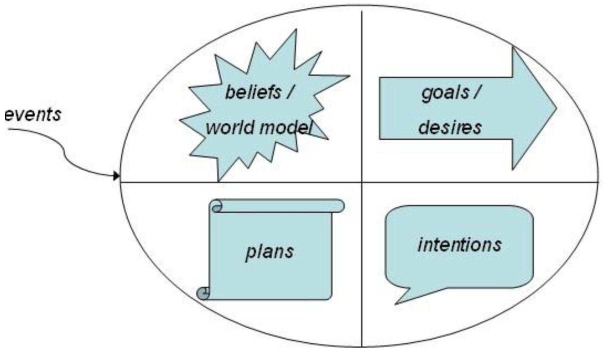


Fig 1. BDI Architecture Structure [4]

Tendency is the target state that the agent likes to achieve and the agent's plans are plans that tell the agent how to behave to achieve his or her intentions. An agent can have a program that triggers a specific activity. In this case, the agent will do exactly the same. Instead, the agent can have a program that can access a particular state. When a factor is committed to performing a specific activity or achieving a particular purpose, it promises to do so. Thus, the BDI will have a set of beliefs that will lead it to satisfy a set of tendencies. In order to achieve this, the BDI agent considers a set of options and chooses one or more of them. These options will now be operational [5,16].

4- Agent Tool Methodology Design

Today, agents usage has considerably increased in computer systems. objectification methods are not able to meet the needs of agent-based software and require essential equipment, which is why software engineering developed from object-oriented to agent-oriented [8]. Openness, high software complexity, distributed data sources and control, and high flexibility are features of systems that the agent technology allows them to generate. The program also has the highest quality and productivity and has the lowest cost [10]. The role of software engineering is to provide models and techniques that facilitate the production and maintenance of the software. with the emergence of a new perspective on the production of software systems, programming languages, software tools and software engineering methods are also appropriate. factors are very similar to objects, but the view and characteristics that distinguish one factor from an object do not allow object-oriented methods to operate on systems based on the factor. For this reason, these systems require software engineering methods based on the unique features of the agents [9,11,12]. Various methodologies have been developed for analyzing and designing agent-based systems [14]. These methodologies can be categorized into two major categories. The first is the methodologies that are based on the development of object-oriented software engineering techniques and their adaptation to the agent's perspective. These methodologies are Gaia, multi-agent system engineering: MaSE [16,17]. MESSAGE [13]. The second category is the methodology that develops engineering knowledge methods.MAS-CommonKADS [18]. CoMoMAS [19] are examples of these methodologies.

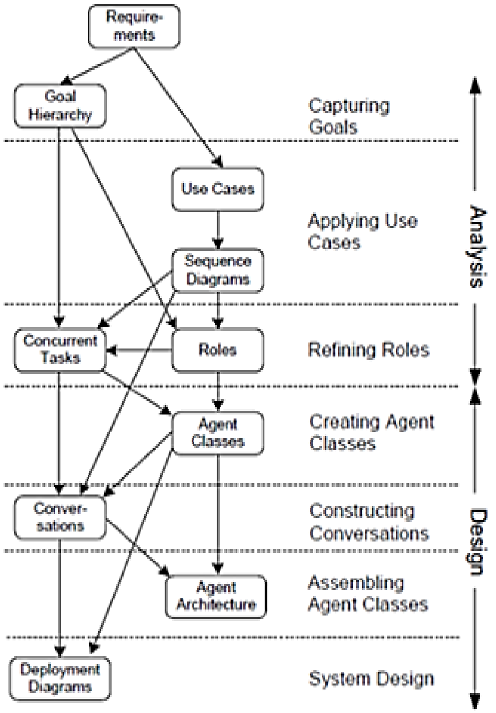


Fig 2. Design and analysis steps in the multi-agent system engineering: MaSE methodology [28]

MaSE is one of the software engineering methodologies for the operating system based on the operating system [10]. Which, in seven steps, passes through the stages of analysis and design, the software engineer implements the problem-to-implementation definition [16, 17]. With the support of the agent-Tool tool that covers all stages of analysis and design and enables the production of code automatically [16, 17]. this methodology is considered one of the most important methodologies in the software engineering of agent-based systems. The MaSE methodology has been used in the fabrication and production of various systems, as well as various studies to address its shortcomings [22, 21,20,17].

5- The MaSE-based Analysis and Design Steps in Agent Tool

5-1- System Analysis Steps

The MaSE methodology consists of two main phases of analysis and design. the MaSE analysis phase consists of three main phases, "setting goals", "defining the applications", and revision of roles in the system [21] ,and the design phase related to topics such as the diagram and classroom conversation [10]. The design and analysis steps in this method are shown in (figure2). [23,22]

5-2- Setting Goals

The first stage of analysis in MaSE is the system's "goal setting" stage, which itself consists of two steps: "detecting targets" and determining the hierarchical chart of objectives. At the stage of diagnosing the objectives, using the definition of the system's requirements, the basic objectives of the system are identified. The "detection of targets" stage begins with the creation of a general scenario of the system. then, based on this scenario, the overall goals of the system are determined. In the stage of classification of goals, the goals set for the system are categorized hierarchically in the form of a tree.

5-3- Determine the Application Cases

The purpose of this step is to identify a set of roles in the system and how they communicate. The "determination of its applications" phase consists of two steps, the creation of "application cases" and the creation of "order charts." In the process of creating "applications", the set of events that may occur in the system is defined. In the process of constructing sequence diagrams, based on the application cases created in the previous step, the order of events between the various roles is displayed using the "order chart".

5-4- Role Reviewing Step

"Role Revision" is the last stage of analysis in MaSE. The purpose of this step is to determine the final roles defined in the system appropriately for design and implementation in a multiprocessor system. At this stage, after the roles are identified, the "tasks" of each role are determined [20]. Each task is displayed using an infinite mode machine. In this case, the "Order Diagram" can be used as the starting point for specifying these state machines, which, in turn, can express the desired states in plain language, provided they are not ambiguous.

5-5- System Design Steps

The models created at the analysis stage are the input of the design stage at MaSE. The operating classes, the effects of each "conversation" on each factor, the internal architecture of the agents, and the "structural arrangement" of the software output system are the design stage [20].

5-6- Create Operating Classes

The first step in designing is the creation of operating classes. At this stage, operating classes are created based on the roles defined in the analysis stage. The agents have different roles in the process of running the system.

5-7- Stage of Dialogue Creation

At this stage, for each connection, a finite state machine is drawn up for each of the agents involved in that connection. Messages in this section are defined based on UML contracts.

5-8- Step Combine Operating Classes

At this stage, the internal architecture of the agents is determined. The internal classes of each type of agent are determined by a set of predefined components and by their combination [24]. The "reactive" agents have the response to the stimulus to the environment. Architectures based on this are based on specific rules. (figure 2)

5-9- Stage System Design

The final stage in MaSE is the system design. At this stage, the actual examples of each agent in that system are displayed in the "structural arrangement" system diagram. The number, type, and location of each sample created in this section are indicated.

6- Analysis and Design of the Proposed Model

The agents are very similar to objects, but the view and characteristics that distinguish one factor from an object do not allow object-oriented methodologies to be appropriate for in-system systems. For this reason, these systems require software engineering methods based on the unique features of the agents [13,12]. So far, various methodologies have been developed for analyzing and designing agent-based systems [14]. Air traffic has steadily increased over the past decade, and ICAO has predicted that annual traffic will double in 2030 compared to 2013. [25]. Air traffic controllers are responsible for ensuring that the flights' are safe and smooth, and if flights' are diverted from the main route, they should be able to quickly return to the main path and to do this, routes between each airport should be maintained [26].

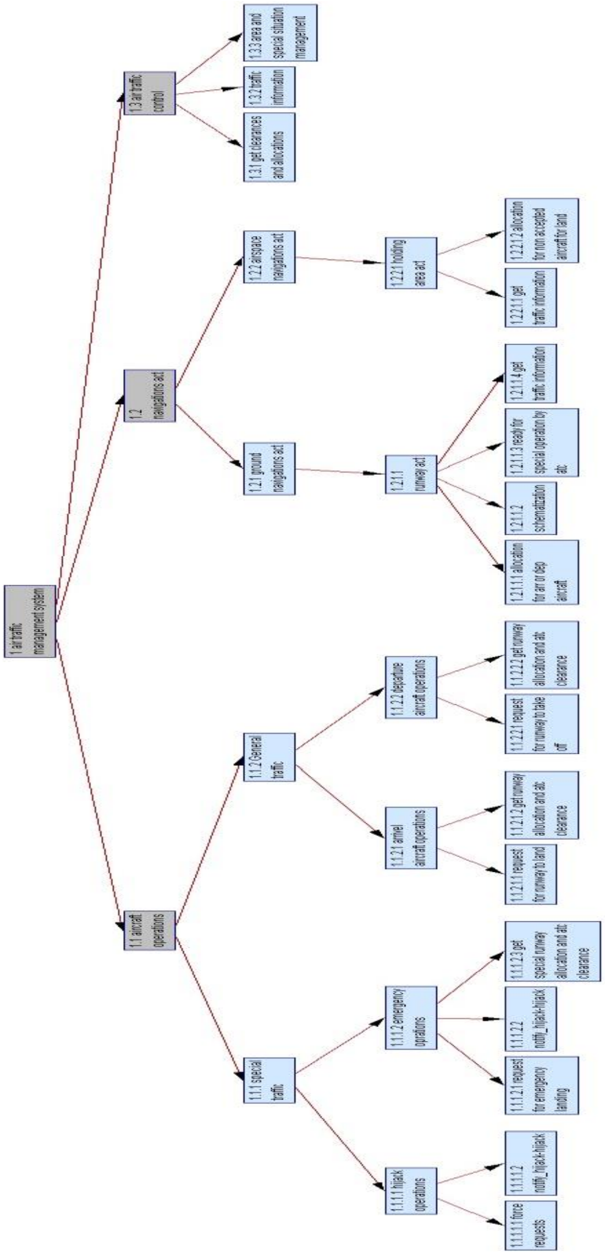

1 air traffic management system
1.3 air traffic control
1.3.3 area and special situation management
1.3.2 traffic information
1.3.1 get clearances and allocations
1.2 navigations act
1.2.2 airspace navigations act
1.2.2.1 holding area act
1.2.2.1.2 allocation for non accepted aircraft for land
1.2.2.1.1 get traffic information
1.2.1 ground navigations act
1.2.1.1 runway act
1.2.1.1.4 get traffic information
1.2.1.1.3 ready for special operation by atc
1.2.1.1.2 schematization
1.2.1.1.1 allocation for arr or dep aircraft
1.1 aircraft operations
1.1.2 General traffic
1.1.2.2 departure aircraft operations
1.1.2.2.2 get runway allocation and atc clearance
1.1.2.2.1 request for runway to take off
1.1.2.1 arrivel aircraft operations
1.1.2.1.2 get runway allocation and atc clearance
1.1.2.1.1 request for runway to land
1.1.1 special traffic
1.1.1.2 emergency oprations
1.1.1.2.3 get special runway allocation and atc clearance
1.1.1.2.2 notify_hijack-hijack
1.1.1.2.1 request for emergency landing
1.1.1.1 hijack operations
1.1.1.1.2 notify_hijack-hijack
1.1.1.1.1 force requests

Fig 3. the hierarchical representation of the objectives of the multi-agent system of air traffic control management

The air traffic system in a network has vertices that represent the geographic location that the airplane can cross over. [27] The goals of the multistage air traffic control system are based on the air traffic control scenario and optimal traffic arrangement and avoidance of any incident. On the other hand, these goals should be in such a way as to satisfy the demands of a system designed in the form of artificial intelligence technology based on multitasking systems, which is to solve the existing problem and not complex the problem. As explained in the previous section, the objectives of our proposed model are categorized according to the MaSE methodology and are presented as a hierarchical chart of objectives [20].

6-1- Hierarchical Chart of Goals

These goals are organized in three general categories, which outline the objectives of the operating groups in the system and the objectives of the system. It should be noted that in determining the objectives, the status of the system should be considered collectively or competitively, because in a multi-competitive system, there is usually a conflict between the goals of the agents and the objectives of the entire conflict system, and the profit of one factor of the system alone contrasts with the total profit of the system, but in collaborative multitasking systems, this is rare. Here, our system is a collaborative multimodal system, since its main objective is aeronautics and air traffic control. The ultimate goal in controlling air traffic control is to optimize the arrangement of air traffic in the airspace and to avoid any disaster. But a multi-agent system has been created from a variety of factors, each with a specific activity and purpose. Each factor, while playing its role in the system, is actually looking for their goal, which we will achieve in the final conclusion of the activities and goals to achieve the main purpose of the system.

Given the three factors in our intelligent proposed model for controlling Air traffic the main purpose of the model is divided into three main subdivisions: aviation factor group goals, airborne mission group goals, atc task force objectives. For example, the objectives of the aircraft group are divided into two sub-divisions of the objectives of ordinary aircraft and the objectives of the special aircraft (at risk). Objectives of the common aircraft can be divided into two sub-groups of the objectives of the incoming aircraft and the objectives of the operating aircraft of the outboard, and then the objectives of the operating aircraft of the aircraft are divided into two categories: the allocation of the band for landing at the airport and the assignment of the band and the request for authorization of the ATC agent landing in the airport is divided according to the priority of the flight. In (figure3), is presented the hierarchical chart of the objectives multi-factor traffic control system.

6-2- System Sequences

In this section, we will detail the five sequence diagrams that result from the objectives and system operation.

6-2-1- Arrangement Diagram of the Incoming Airplane Landing Operation

The incoming air carrier will send its flight information to the operating tower of the flight tower to assign the landing gear, as well as the incoming airline agents in the aircraft operating group to send their flight information. Upon receipt by the ATC agent, the agent sends a request for the occupation of the airport landing bands to the carrier's airborne commander, and then the agent sends the request to determine the best condition for the sub-category landing bands which it sends itself. After identifying the best landing route, it sends its specifications to the ATC agent, and the agent sends the updated timetable to the ATC agent, after sending the assigned command to the target band. The ATC agent sends permission to assign the specified band to the intended aircraft. (figure. 4)

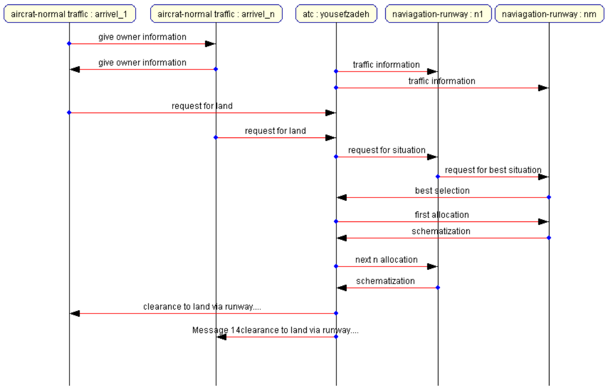


Fig 4. Arrival chart of the incoming airplane landing operation

6-2-2- Arrangement Diagram Outboard Aircraft Lifting Operation

The outboard agent sends a request to the flying tower to allocate the exit channel to the flight attendant. Also, the airliner sends flight information to the aircraft operating group. Upon receipt by the ATC agent, this agent sends a request for the occupation time of the airport take-off bands from the airport's airborne group, and then each of them reports their status (declaring readiness or being busy) to the agent ATC sends.

The ATC agent sends the assignment command to the desired band. The operating agent sends its updated timetable to the ATC agent. This agent sends the necessary permission to assign the specified band to the agent (aircraft) concerned. Of course, it should be noted that the ATC agent sends traffic information to the navigation agents in order to accelerate the system's performance. (figure.5)

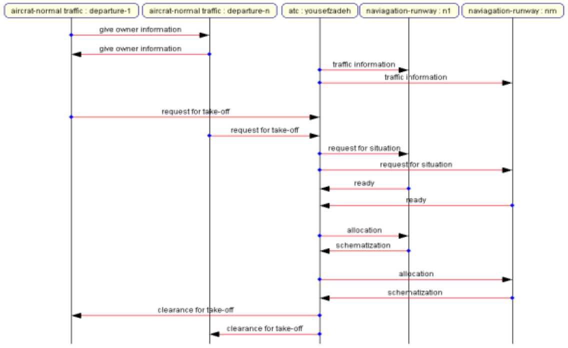


Fig 5. Arrangement diagram of the lifting operation of the exits

6-2-3- Arrangement Diagram of the Operation of Entering the Aircraft into the Holding Area

The incoming air carrier will send its flight to the tower of the flight tower in order to allocate the landing gear. Upon receipt by ATC, this agent sends an application to the occupant of the airport landing bands to send the airline flight officer headquarters. The director of the navigational group after the survey and during the steps mentioned above, sends an answer to the fact that all the airport landing bands are busy with the ATC agent. The ATC agent then sends the agent (aircraft) a non-landing message. The ATC agent sends the flight traffic information to the navigation operator of the landing area, and then issues the decree to the operating agent (aircraft) and the license to start the hiding area project. (figure. 6)

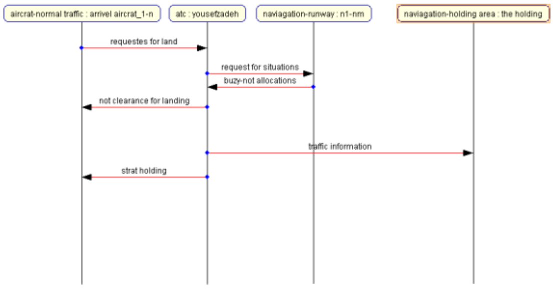

aircrat-normal traffic : arrivel aircrat_1-n
atc : yousefzadeh
naviagation-runway : n1-nm
naviagation-holding area : the holding
requestes for land
request for situations
buzy-not allocations
not clearance for landing
traffic information
strat holding

Fig 6. Airplane Entry Operations to the Holding Area

6-2-4- Flow Diagram Diagrams in Emergency Situations

The agent also receives flight information from other aircraft agents. The ATC agent sends the traffic status to the airborne agents, and then sends a message to the emergency agent and alert mode to the navigation agent. After that, the ATC agent gives all the planes a decree decommissioning from the space specified for the emergency landing. (figure.7)

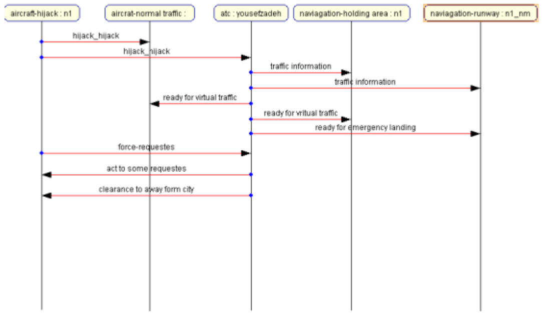


Fig 7. Operations Diagram of Emergency Operations

6-2-5- Arrangements for Aircraft Operations During the Hijacking

The airplane hijacking agent's airborne agent will send the flight agent to the flying tower and other flying agents (onboard aircraft). The ATC agent sends the traffic status to airborne agents, and then sends a hailing message and requesting an alert status to the agents, including the landing gear and the holding area. The ATC agent then orders all planes to be ready for artificial traffic creation to limit the abducted airplane's space. The abducted aircraft agent sends requests to the ATC agent, and the ATC agent tries to meet their apparent needs by helping them from the virtual traffic mode to the abducted aircraft, sending messages that control and dispose of them from the urban and rural spaces. (figure.8)

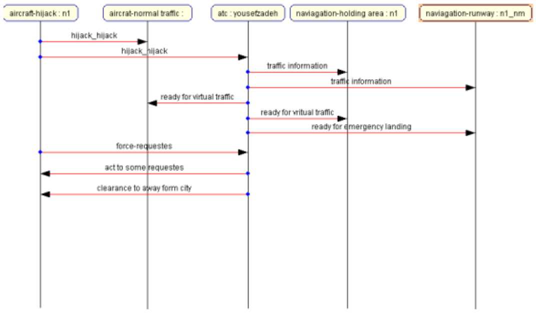


Fig 8. Arrivals of airplane operations during the hijacking

6-3- Graph of the Expression of Roles and Tasks

As described in Section 6-2, the "review roles" stage is the most important stage in MaSE's analysis. The goal of this step is to determine the roles defined in the system appropriately for design and implementation in a multi-purpose system. In the proposed model for air traffic management multifunctional system, according to classified objectives, several roles such as normal-type aircraft (inertial, outbound, local), abducted aircraft, emergency planes, flight controllers, flight bands (landing, take off), stops and so on. At this stage, after the roles are identified, the "tasks" of each role are determined. Given the sequence diagrams described in the previous section, which are described in detail, after the determination of the roles, it is time to determine the tasks. In our proposed model for the air traffic control multitasking system, tasks are divided into two categories: data processing (traffic and airport information) and flight instructions (requests, commands, issuance of licenses, etc.). "The role of flight attendance" also has a supervisory and supervisory role. Each of these roles includes and follows the specific objectives described in the hierarchical objectives. (figure.9)

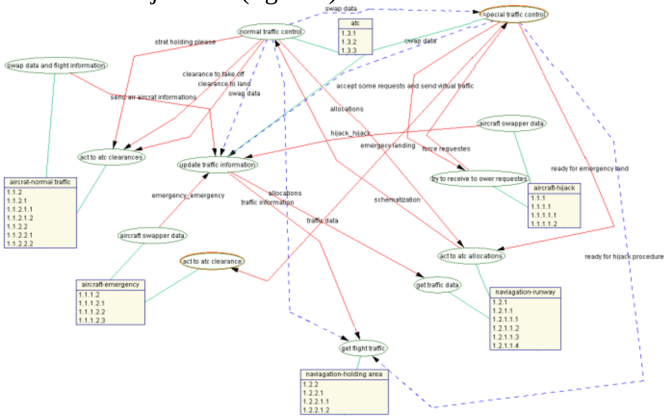


Fig 9. Graphs of Expressions of Roles and Function

6-4- Charts for Designing Operating Classes and their Combining Expressions

At this stage, operating classes are created based on the roles defined in the analysis stage. Factors play different roles during the process of system execution. In our proposed model for air traffic control management, based on the criteria mentioned, three functional class groups are considered, which include the operating group of flight care, the aircraft operating group, and the aeronautical operating group. The factor group is the facilitator of the operation mini the "reactive" factors have the characteristics of the stimulus- response to the environment. Architectures based on this are based on specific rules. (Fig.10)

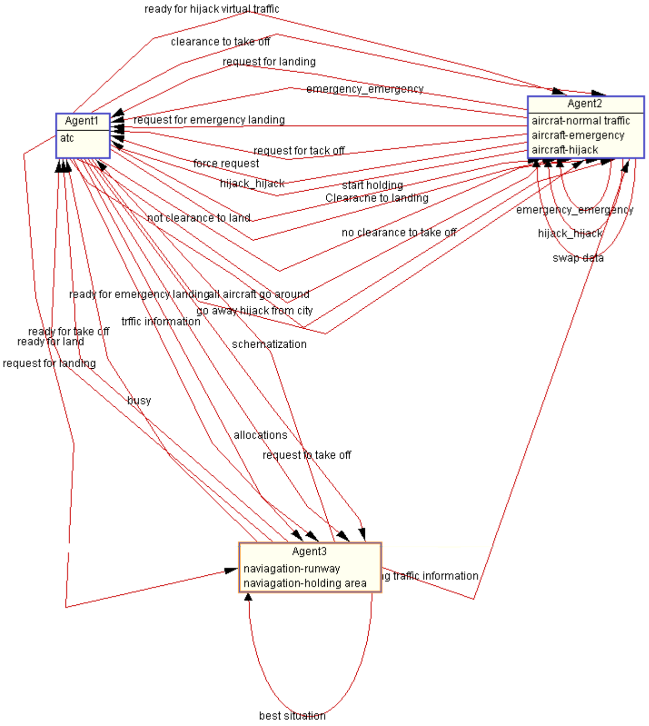


Fig 10. Design Diagrams of the Operating Classes and Expressions of Their Combinations

7- Discusion and Results

The goals have been regulated in the form of three sub-collection which are indications of the agent groups' goals in system and system goals. To determine the goals ,the situation of the system should be cooperative and or competitive, because ,in competitive multi-agent systems, there is a contradiction between agents goals and total goals system and the interest of an agent out of system contradicts with its total interest, but in a multi-agent system this problem is rare .

Here, our system has been determined with the kind of cooperative multi-agent systems, because our main goal is in the direction of stall speed and secure air traffic control.

On the basis of three agent bands in the suggested intelligent model to control air traffic, the major goal of the model is divided into three sub-classifications:

- aircraft agent group goals
- air navigational agent group goals
- ATC agent group goals

For example, aircraft agent group goal with two sub-structure:

- normal aircraft agent goals
- special agents goals(in risk)

also, normal agents goals are as follows:

- arrival (aircraft)agent goals
- departure(aircraft)agent goal

agents goals arrival (aircraft) are two sub-groups:

- request of allocating a runway to land in the airport

The runway allocation and request of license from ATC agent to land in the airport on the basis of flight priorities .Table 1 shows an example of flight priority rules derived from intelligent operating systems that help evaluate multi-agent consulting systems in air traffic management. Due to the increase of conditions and their combination, we will present some of the rules as an example in this table.

Table 1. flight rules in agent-base system

1- If aircraft arrival or departure from runway then aircraft movement into the wind
2- arrival aircraft has high preference than departure aircraft
3- If aircraft arrival into the holding area then { If aircraft input from part 1 then aircraft turn left and proceed to procedure "a" of holding area If aircraft input from part 2 then aircraft turn right till radial 30 degree and proceed to procedure "b" of holding area If aircraft input from part 3 then aircraft turn right and proceed to procedure "c" of holding area}
4- in arrival or departure if medium aircraft behind heavy aircraft then 2 minute (minima time separation) shall be applied
5- in arrival or departure if light aircraft behind medium or heavy aircraft then 3 minute (minima time separation) shall be applied
6- fighter aircraft same as light aircraft in priority.
7- hijack and emergncy must be away from city.
8- if(aircraft[i][3].compareTo("arr")==0;
9- if(aircraft[i][1].compareTo("emergency")==0 aircraf[i][6]=5;
10- if(aircraft[i][1].compareTo("air-amb")==0 aircraf[i][6]=4;
11- if(aircraft[i][1].compareTo("sar")==0 aircraf[i][6]=3;
12- if(aircraft[i][1].compareTo("vip")==0 aircraf[i][6]=2;
14- if(aircraft[i][3].compareTo("dep")==0;
13- if(aircraft[i][1].compareTo("acft")==0 aircraf[i][6]=1;
15- if(aircraft[i][1].compareTo("scramble")==0 aircraf[i][6]=5;
16- if(aircraft[i][1].compareTo("air-amb")==0 aircraf[i][6]=4;
17- if(aircraft[i][1].compareTo("sm")==0 aircraf[i][6]=3;
18- if(aircraft[i][1].compareTo("sar")==0 aircraf[i][6]=2;
19- if(aircraft[i][1].compareTo("vip")==0 aircraf[i][6]=1;
20- if(aircraft[i][1].compareTo("normal")==0 aircraf[i][6]=0;
21- if(aircraft[i][2].compareTo("heavy")==0 aircraf[i][7]=0.3;
22- if(aircraft[i][2].compareTo("medium")==0 aircraf[i][7]=0.2;
23- if(aircraft[i][2].compareTo("light")==0 aircraf[i][7]=0.1;

8- Conclusion

The use of intelligent agent-based systems in air traffic management is one of the important issues in the field of artificial intelligence knowledge and modern aeronautical sciences. In this paper, we tried to design a new model for a multi-agent traffic management system. In this model, the structure of the agents and their interactions are selected based on the BDI model. Then the design of the multi-agent system and operating classes under the MaSE methodology was done using the agent tools. Due to the fact that there was a distinction between emergency and airlift control and other flights', the results of the system test showed a relative improvement in the parameters being evaluated. In further studies, to use deep learning networks methods such as recurrent neural networks (RNN) and long-short term memory(LSTM) in agent-based air traffic model design to increase the ATM accuracy

References

[1] F. Zambonelli, N. R. Jennings, and M. Wooldridge, “Organisational abstractions for the analysis and design of multi-agent systems,” in International Workshop on Agent-Oriented Software Engineering, 2000, pp. 235–251.

[2] ICAO, “Manual on Air Traffic Management System Requirements (doc. 9882).” p. 67, 2008

[3] V. N. Drozdov, V. A. Kim, and L. B. Lazebnik, “[Modern approach to the prevention and treatment of NSAID-gastropathy].,” Ėksperimental’naia i klinicheskaia gastroėnterologiia = Experimental & clinical gastroenterology, no. 2. pp. 106–110, 2011.

[4] S. Zilberstein, “Book Review: ‘Multiagent Systems: a Modern Approach To Distributed Artificial Intelligence’, Gerhard Weiss,” International Journal of Computational Intelligence and Applications, vol. 01, no. 03. pp. 331–334, 2001, doi: 10.1142/s1469026801000159.

[5] K. S. Obhan, “Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence By Gerhard Weiss.

[6] V. Smirnova, “Multi-agent System for Distributed Data Fusion in Peer-to-Peer Environment,” Masters Study, 2002.

[7] M. Niazi and A. Hussain, “Agent-based computing from multi-agent systems to agent-based models: A visual survey,” Scientometrics, vol. 89, no. 2, pp. 479–499, 2011, doi: 10.1007/s11192-011-0468-9.

[8] B. Henderson-sellers, “Agent-based Software Development Methodologies What is an Agent?” 1997.

[9] N. R. Jennings, “Agent-oriented software engineering,” in European Workshop on Modelling Autonomous Agents in a Multi-Agent World, 1999, pp. 1–7.

[10] E. Amini and F. S. Gharehchopogh, “Analysis and Design of New Secure Dynamic Structure for Increasing Modularity of ERP Systems.”

[11] N. R. Jennings, “On agent-based software engineering,” Artif. Intell., vol. 117, no. 2, pp. 277–296, 2000.

[12] F. Zambonelli, N. R. Jennings, and M. Wooldridge, “Developing multiagent systems: The Gaia methodology,” ACM Trans. Softw. Eng. Methodol., vol. 12, no. 3, pp. 317–370, 2003.

[13] R. Evans et al., “MESSAGE: Methodology for engineering systems of software agents,” Eurescom, Edin, no. September 2001. pp. 223–907, 2001.

[14] C. A. Iglesias, M. Garijo, and J. C. González, “A survey of agent-oriented methodologies,” in International Workshop on Agent Theories, Architectures, and Languages, 1998, pp. 317–330.

[15] A. M. de Souza, C. A. R. L. Brennand, R. S. Yokoyama, E. A. Donato, E. R. M. Madeira, and L. A. Villas, “Traffic management systems: A classification, review, challenges, and future perspectives,” Int. J. Distrib. Sens. Networks, vol. 13, no. 4, 2017, doi: 10.1177/1550147716683612.

[16] M. O. Khozium, “Multi-agent system overview: architectural designing using practical approach,” Int. J. Comput. Technol., vol. 5, no. 2, pp. 85–93, 2013.

[17] N. Glaser, State of the Art: Knowledge Acquisition and Modelling. 2002.

[18] G. Schreiber, B. Wielinga, R. de Hoog, H. Akkermans, and W. de Velde, “CommonKADS: A comprehensive methodology for KBS development,” IEEE Expert, vol. 9, no. 6, pp. 28–37, 1994.

[19] N. Glaser, “Contribution to knowledge acquisition and modeling in a multi-agent framework PhD Thesis,” UHP Nancy I, pp. 87–127, 1996.

[20] R. H. Bordini et al., “A survey of programming languages and platforms for multi-agent systems,” Inform., vol. 30, no. 1, pp. 33–44, 2006.

[21] L. Bianco, P. Dell’Olmo, and A. R. Odoni, New concepts and methods in air traffic management. Springer Science & Business Media, 2013.

[22] International Civil Aviation Organization, “Global Air Navigation Plan for CNS / ATM Systems,” Training, no. Doc 7300. 2002.

[23] J. C. Hill, F. R. Johnson, J. K. Archibald, R. L. Frost, and W. C. Stirling, “A cooperative multi-agent approach to free flight,” Proceedings of the International Conference on Autonomous Agents. pp. 1205–1212, 2005, doi: 10.1145/1082473.1082637.

[24] D. Robinson, “A Component Based Approach to Agent Specification, Department of electrical and computer engineering Air Force Institute of Technology,” M. Sc. Thesis, 2000.

[25] R. Breil et al., “Multi-agent Systems for Air Traffic Conflicts Resolution by Local Speed Regulation To cite this version: HAL Id: hal-01342623 Multi-agent Systems for Air Tra ffi c Conflicts Resolution by Local Speed Regulation,” 2016.

[26] J. M. Canino, J. B. Portas, J. M. Molina, J. Garc\’\ia, and T. Magister, “A multi-agent approach for designing next generation of air traffic systems ,” in Advances in Air Navigation Services , InTech, 2012.

[27] D. R. Isaacson, A V Sadovsky, and D. Davis, “Tactical scheduling for precision air traffic operations: Past research and current problems,” J. Aerosp. Inf. Syst., vol. 11, no. 4, pp. 234–257, 2014.

[28] Self, Athie L., and Scott A. DeLoach. "Designing and specifying mobility within the multiagent systems engineering ethodology." Proceedings of the 2003 ACM symposium on Applied computing. 2003.

[29] Molina, Martin, Sergio Carrasco, and Jorge Martin. "Agent-based modeling and simulation for the design of the future european air traffic management system: The experience of

cassiopeia." International Conference on Practical Applications of Agents and Multi-Agent Systems. Springer, Cham, 2014.

[30] Gore, Brian F. "Man–machine integration design and analysis system (MIDAS) v5: Augmentations, motivations, and directions for aeronautics applications." Human modelling in assisted transportation. Springer, Milano, 2011. 43-54.

[32] Šišlák, David, et al. "AGENTFLY: Towards multi-agent technology in free flight air traffic control." Defence industry applications of autonomous agents and multi-agent systems. Birkhäuser Basel, 2007. 73-96.

[33] Molina, Martin, Sergio Carrasco, and Jorge Martin. "Agent-based modeling and simulation for the design of the future european air traffic management system: The experience of cassiopeia." International Conference on Practical Applications of Agents and Multi-Agent Systems. Springer, Cham, 2014.

[34] Becker-Asano, Christian, et al. "A multi-agent system based on unity 4 for virtual perception and wayfinding." Transportation Research Procedia 2 (2014): 452-455.

[35] Bazzan, Ana LC, and Franziska Klügl. Multi-agent systems for traffic and transportation engineering. IGI Global, 2009.

[36] Breil, Romaric, et al. "Multi-agent systems to help managing air traffic structure." Journal of Aerospace Operations 5.1-2 (2017): 119-148.

[37] Abar, Sameera, et al. "Agent Based Modelling and Simulation tools: A review of the state-of-art software." Computer Science Review 24 (2017): 13-33.

[38] Gurtner, Gérald, et al. "An empirically grounded agent based simulator for the air traffic management in the SESAR scenario." Journal of Air Transport Management 59 (2017): 26-43.

[39] Bouarfa, Soufiane, Jasper Müller, and Henk Blom. "Evaluation of a Multi-Agent System approach to airline disruption management." Journal of Air Transport Management 71 (2018): 108-118

Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 37- 48

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Training and Learning Swarm Intelligence Algorithm (TLSIA) for Selecting the Optimal Cluster Head in Wireless Sensor Networks

Ali Sedighimanesh[1], Hessam Zandhessami[1*], Mahmood Alborzi[1], Mohammadsadegh Khayyatian[2]

[1].Department of Management and Economics, Science and Research branch, Islamic Azad University, Tehran, Iran
[2].Institute for science and technology studies, shahid Beheshti university, Tehran, Iran

Received: 28 Nov 2020 / Revised: 29 Apr 2021/ Accepted: 29 May 2021
DOI: https://doi.org/10.52547/jist.15638.10.37.37

Abstract

Background: Wireless sensor networks include a set of non-rechargeable sensor nodes that interact for particular purposes. Since the sensors are non-rechargeable, one of the most important challenges of the wireless sensor network is the optimal use of the energy of sensors. The selection of the appropriate cluster heads for clustering and hierarchical routing is effective in enhancing the performance and reducing the energy consumption of sensors. Aim: Clustering sensors in different groups is one way to reduce the energy consumption of sensor nodes. In the clustering process, selecting the appropriate sensor nodes for clustering plays an important role in clustering. The use of multistep routes to transmit the data collected by the cluster heads also has a key role in the cluster head energy consumption. Multistep routing uses less energy to send information.
Methods: In this paper, after distributing the sensor nodes in the environment, we use a Teaching-Learning-Based Optimization (TLBO) algorithm to select the appropriate cluster heads from the existing sensor nodes. The teaching-learning philosophy has been inspired by a classroom and imitates the effect of a teacher on learner output. After collecting the data of each cluster to send the information to the sink, the cluster heads use the Tabu Search (TS) algorithm and determine the subsequent step for the transmission of information. Findings: The simulation results indicate that the protocol proposed in this research (TLSIA) has a higher last node dead than the LEACH algorithm by 75%, ASLPR algorithm by 25%, and COARP algorithm by 10%.
Conclusion: Given the limited energy of the sensors and the non-rechargeability of the batteries, the use of swarm intelligence algorithms in WSNs can decrease the energy consumption of sensor nodes and, eventually, increase the WSN lifetime.

Keywords: Hierarchical Routing; TLBO Algorithm; TS Algorithm; Wireless Sensor Network.

1- Introduction

The wireless sensor network consists of several non-rechargeable sensor nodes applied for particular purposes [1]. One of the most important issues and challenges related to wireless sensor networks is the use of methods to reduce the energy consumption of sensor nodes. One of the methods is the clustering of the sensor nodes; instead of the sensor nodes consuming a great deal of energy and transmitting the data directly to the sink, they fall into a group called the cluster and send the data to the cluster head, and the cluster heads are required to transmit the data, thus consuming less energy of the sensor nodes and extending the network's lifetime [2]. Cluster heads can either send the received data directly to the sink or work together to send the data to the sink in a hierarchical routing process. In general, transmitting data hierarchically reduces the energy consumption of cluster heads farther from the sink [3],[4].
The process of selecting cluster heads from available sensors and the routing between clusters to transmit data to the sink are of the optimization issues; therefore, the use of optimization algorithms has an effective role in the proper performance of these two processes, and ultimately, the efficiency of the wireless sensor network [5],[6]. Teaching-Learning-Based Optimization (TLBO) algorithm is one of the modern intelligent optimization algorithms implemented in two stages (phases) and can lead to optimization through being inspired by the learning and teaching process. In the teaching phase, the best member of the community is selected as the teacher and directs the

🖂 **Corresponding Author**
Zandhessami@srbiau.ac.ir

average population towards himself/herself; this is similar to what a teacher does in the real world. In the learning phase, the people in the population work together to increase their knowledge, and it is similar to what happens in the company of friends and classmates [7].

The Tabu Search (TS) [8] algorithm is also one of the most powerful algorithms for solving optimization problems, especially graph-based and combinatorial optimization problems. The TS algorithm applies a list named the taboo list, which has been designed to prevent the algorithm from falling at the local optimal point. In summary, TS starts from a point or solution and searches for neighbors around that point, chooses the best neighbor and moves to that point, and continues this search until a stopping criterion to be satisfied. The optimal point is reported at the end of the search.

In the present article, the TLBO swarm intelligence algorithm is applied to select the appropriate cluster heads from the available sensor nodes. Once the cluster heads are identified, the members of each cluster become the member of the nearest cluster head and send the data to their cluster heads. The cluster heads receive data from their members and process and aggregate them subsequently. Then, the TS algorithm is used to transmit data to the sink by cluster heads until the best routes are formed for sending data, which reduces the energy consumption of cluster heads to transfer data. The rest of the article is structured as follows. Section 2 presents the previous work. Section 3 addresses the Proposed algorithm. Section 4 discusses the findings of the article. In Section 5, the authors present open problems for wireless sensor networks, and also the results are presented.

2- Previous Works

In this research, we will address several routing protocols that have attracted interest in recent years, namely the following: LEACH, ASLPR, and COARP[9][10].

2-1- Low- Energy Adaptive Clustering Hierarchy (LEACH)

In the LEACH protocol [11], there is a probability P for each sensor to be a cluster head (CH) in every round. In other words, LEACH creates groups using a distributed algorithm, in which the sensors automatically decide to become a cluster head and there is no centralized control. Each sensor can be a cluster head only once in 1/P consecutive rounds. First, each sensor makes a decision with a probability of P to become a cluster head. The cluster head roles changes in rounds between the group nodes, and this is to create an equilibrium in the energy consumption distribution. One can divide the performance of LEACH in each round into two phases. These phases are the setup and steady-state phases. A random number between 0 and 1 is chosen by every sensor in the setup phase. If that number is smaller than T(n), the sensor n becomes a CH for that round. The value of T(n) is computed based on (1), where P is the tendency of the sensor to be a node, and r represents the round number. Moreover, G denotes the set of all sensors that have not been chosen as a cluster head during the last 1/P rounds.

$$T_{(n)} = \{\frac{P}{1 - P \times [r\, mod\left(\frac{1}{P}\right)]} \quad if\ n \\ \in G\ 0 \qquad Otherwise \tag{1}$$

After the cluster heads are selected, they are announced to all the sensors in the network as cluster heads. When non-cluster head sensor receives an announcement from the cluster heads, it selects the cluster head closest in terms of communication.

2-2- Application- Specific Low Power Routing (ASLPR) protocol

The ASLPR protocol [12] collects specific pieces of information, such as remaining energy, distance from the base station, and distance between the CHs and sensor node, to select the cluster head nodes. Then, each node selects a random number between zero and 1. If the random number selected by a node is less than T_{ASLPR} in (2), this node is converted to a cluster head.

$$T_{ASLPR} = \begin{cases} Z(n) & if\ E(n) \geq t_1 \times \frac{1}{N}\sum_{i=1}^{N} E(i) \\ 0 & if\ E(n) < t_1 \times \frac{1}{N}\sum_{i=1}^{N} E(i) \end{cases} \tag{2}$$

$$Z(n) = \alpha_1 T_1(n) + \alpha_2 T_2(n) + \alpha_3 T_3(n) + \alpha_4 T_4(n) \tag{3}$$

In the above relationships, N represents the total number of live nodes in the current round, and E(n) equals the n remaining nodes.

In (3), $T_1(n)$ denotes the sub-threshold of the node energy, and α_1 refers to the weight of this sub-threshold. Moreover, $T_2(n)$ represents the sub-threshold for the distance between the nodes and the base station, and α_2 denotes the weight of this sub-threshold. In addition, $T_3(n)$ is the sub-threshold for the distance between the node and the cluster head, and α_3 refers to the weight of this sub-threshold. The sub-threshold $T_4(n)$ denotes the number of rounds where a node has been the cluster head, and α_4 represents the weight of this sub-threshold. Then, the cluster head nodes announce their existence to all the nodes in the network by issuing a message. After receiving this message from different cluster heads, the regular (non-cluster head) nodes select the closest cluster head to join. In this protocol, genetic algorithm (GA) combined with the simulated annealing (SA) algorithm has been used to optimize the special parameters utilized for determining the threshold for application-specific cluster heads. The objective functions of the GA and SA algorithms in this protocol are defined as follows:

$$Maximize: fitness = W_1 \times FND \times W_2 \times HND + W_3 \times LND \qquad (4)$$

$$0 \leq \alpha_k \leq 1 \qquad (k = 1,2,3,4), \quad \sum_{k=1}^{4} \alpha_k = 1 \qquad (5)$$

$$0 \leq t_s \leq 2 \qquad (k = 1,2,3,4), \quad t_1 \leq t_2 \qquad (6)$$

$$0 \leq W_u \leq 1 \qquad (k = 1,2,3), \quad \sum_{k=1}^{3} W_u = 1 \qquad (7)$$

In the above relationships, W_1, W_2, and W_3 denote the weights of the First Node Dead (FND), Half Node Dead (HND), and Last Node Dead (LND), respectively. The ranges of the mentioned weights are between 0 and 1, depending on the application, such that their sum equals 1 according to (7). Moreover, t_s refers to the sub-threshold values in (3), and α_k in (4) represents the weight of the sub-threshold in (3).

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

In COARP [13], measurements to determine the CHs are performed within a centralized control system. The model of the network is a single-step model where the CHs communicate directly with the base station. During every round, the base station is aware of the position and energy level of the nodes in the network. During each round, every node sense and gathers the surrounding data. Then, it processes the data and sends it to the cluster head in a data packet form. The COARP clustering method involves the following steps: (1) the start-up phase, which involves determining the cluster head and creating the cluster, (2) the register phase, which involves creating a data scheduling and transmission plan. In CAORP, the CHs are accurately chosen by the cuckoo algorithm in the base station. Then, the cluster creation process and the register phase are performed. Every CH receives the information relating to all the nodes belonging to its own cluster. Then, it sends the received information to the base station in the form of a packet.

3- Proposed Algorithm

The appropriate selection of cluster heads from the available sensor nodes is one of the methods that lead to the reduction of the energy consumption of sensor nodes and cluster heads. Besides, the data transmission in a hierarchical manner instead of the one-step method highly affects the reduction of the energy consumption of sensors since the farther apart the two nodes are, the more energy they have to expend for data transmission. Therefore, selecting the appropriate cluster head from the available nodes and the hierarchical routing can lead to the reduction of the energy consumption of the sensor nodes, which will increase the lifetime of the wireless sensor network. For this purpose, there are various methods; the application of optimization methods for solving such problems will enhance decision-making and increase the efficiency of algorithms.

The proposed algorithm described in three sections: sensor node distribution, clustering process, routing. In the sensor node distribution section, the authors explain how to distribute the nodes in the simulation environment. In the clustering section, there is an attempt to classify sensor nodes into different clusters for the purpose of reducing energy consumption. For this purpose, a swarm intelligence algorithm called TLBO is employed to select the optimal cluster heads from the sensor nodes. In the routing section, the objective is to apply the best routes to transmit data hierarchically with less energy consumption; hence, the TS algorithm is used to choose the best route for data transmission. In the following, the authors will explain these steps step by step. The general algorithm of the proposed algorithm is as follows.

TLSIA Algorithm

```
Select nodes in sensing area for clustering
CHs= TLBO
For i=1: number of nodes
      If node(i) is in sensing area && node(i) is normal node
            node(i) joins to nearest CH
      end if
end for
Routing to send cluster head information
Route= TS
For i= Cluster heads
      CH(i) joins to route;
end for
```

3-1- Node Distribution and Sink Location

During the simulation, the sensor nodes are randomly distributed in an environment. Then, the location of the sink is determined, which is usually outside the environment.

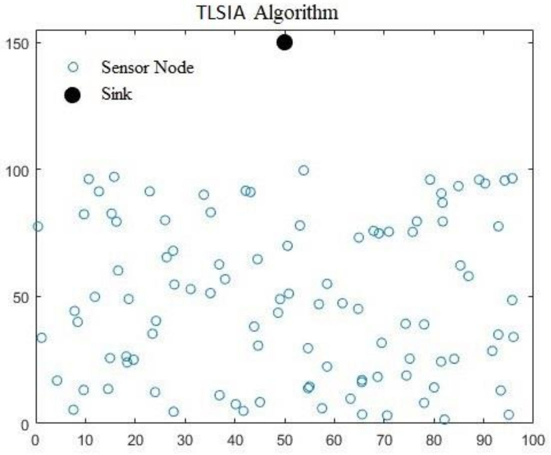


Fig. 1. Random distribution of the nodes in the environment

3-2- Cluster Head Selection

The process of choosing the optimal cluster heads from between the sensors in the network is performed using the Teaching-Learning-Based Optimization (TLBO) algorithm. The teaching-learning philosophy has been inspired by a classroom and imitates a teacher's effect on the learner output. Similar to other swarm intelligence algorithms, the TLBO algorithm is a population-based evolutionary optimization algorithm and consists of a teaching phase and a learner phase.

In the teaching phase, the teacher has the main role and attempts to transfer their knowledge to all the learners in the classroom to increase the average score. The average result of the learners and the improvement in results completely depends on the teacher. In each step, the best learner in the population is selected as the teacher, and, accordingly, the cost function and the average position for improving the position of the learners are computed.

In the learning phase, the learners increase their knowledge either via the teacher or via interacting with each other. The main difference between the teaching and learning phases is that in the teaching phase, the teacher transfers the knowledge to the learners, but in the learning phase, the learners gain knowledge from the teacher and by communicating with each other. In population-based optimization methods, a population has a set of members, each of which has a number of variables. Every member of the population is a solution to the optimization problem. In this paper, we first form an initial population consisting of a number of members, named learners, to determine the cluster head. Each learner includes 2 variables: Position, which consists of a string of variables, and cost. The figure below shows an overview of a population.

Learner 01	Position	Node 01	Node 02	Node 03	….	Node (n-1)	Node (n)
	Cost						
Learner 02	Position	Node 01	Node 02	Node 03	….	Node (n-1)	Node (n)
	Cost						

.
.
.

Learner 0N	Position	Node 01	Node 02	Node 03	….	Node (n-1)	Node (n)
	Cost						

Fig. 2. Overview of a population

First, the variables inside the position are given a random value between 0 and 1 ($0 \leq \text{Position (i)} \leq 1$). The most important issue in optimization algorithms is how to determine the cost for the learners in the population. In this paper, the cost is equal to (8):

$$\text{Cost} = \text{Sum}(\text{Alpha} \times \text{RE}(x), \text{Beta} \times \text{Density}(x), \text{Gamma} \times \text{Centerality}(x)) \quad (8)$$

In the above formula, x is the variable inside the population member, RE is the remaining energy of each variable, density is the ratio of the number of neighbors to the total number of nodes, centrality is the sum of distances of the nodes from the neighbors, Beta= -0.3, Alpha= -0.5, and Gamma=0.2.

In the TLBO method, every member of the population is considered a learner. In every iteration of the TLBO algorithm, we select the member with the lowest cost between the population members as the best member of the population. Then, we sort the variables inside the selected member in descending order and select 10% of these variables as the optimal cluster head. For example, if after the end of the maximum iteration of the algorithm, the output is as follows:

		01	02	03		n-1	n
Learner 01	Position	0.36	0.47	0.25	….	0.12	0.22
	Cost= -1.25						
Learner 02	Position	0.26	0.17	0.45	….	0.32	0.52
	Cost=-1.35						

.
.
.

Learner N	Position	0.14	0.32	0.54	….	0.33	0.63
	Cost=-1.05						

Learner 02 is selected as the best member of the population; hence, the variables inside this member are sorted in descending order, and 10% of them are considered as the cluster head.

In implementing the TLBO algorithm, 3 values have a vital role in the optimal performance of the algorithm: (1) initialization of the learners, (2) updating of the teaching phase, and (3) updating of the learning phase.

Learner initialization: In this method, we first create a random population and calculate the second population from the first using (9). Subsequently, we combine the 2 populations and compute and sort the costs of the learners. Then, we select from the learners with less cost a number equal to the learner members of the population[14], [15].

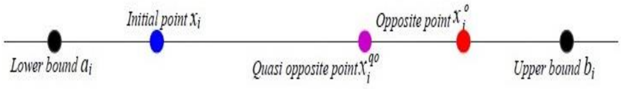


Fig. 3. Opposition-based learning and quasi-oppositional learning[15].

$$x_i^o = a_i + b_i - x_i \tag{9}$$

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

Teaching phase: In the teaching phase, the learners increase their knowledge via learning from the difference between the class average and the teacher. The update mechanism for the ith learner has been expressed as follows:

$$newX_i = X_i + rand.(Teacher - TF.Meam) \tag{11}$$

$$Mean = \frac{1}{NP}\sum_{i-1}^{NP} X_i \tag{12}$$

$newX_i$ is the learner's new state, X_i is the ith learner, Teacher is the learner with the best fitness, NP denotes the number of learners present in the population, and TF is a teaching factor that determines the value of the average that must be changed. Also, rand is a random vector the element of which is a random number in the range [0, 1].

Learning phase: During the learning phase, the learners also increase their knowledge interactively. The update mechanism for the ith learner has been expressed as follows:

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

where newX$_i$ is the ith learner's position, X_k represents the learners chosen randomly from the class, and $f(X_i)$ and $f(X_k)$ respectively denote the fitness values of the learners X_i and X_k. In addition, rand denotes a random vector in the [0, 1] range.

TLSIA Clustering Algorithm

```
Initialize learners;
Evaluate learners;
For all learners
For i=each dimension
    $x_i^o = a_i + b_i - x_i$
    $x_i^{qo} = \frac{a_i+b_i}{2} + rand.\left(x_i^o - \frac{a_i+b_i}{2}\right)$
End_For
End_For
Combine first population and Quasi-opposite population;
Select best learners as new population;
Xteacher=best learner;
Xmean=average of learners;
While (stopping condition is not met)
        For i=all learners
        TF = round (1 + rand (0,1));
        Xnewi=Xi+rand*(Xteacher-TF*Xmean);
        End_For
        Evaluate new learners;
        If new learner is better than old one
            Xi=Xnewi;
        End_If
        For i=all learners
            Randomly select another learner which is different from i (Xk);
            If Xi is better than Xk
                Xnewi=Xi+rand*(Xi-Xk);
            Else
                Xnewi=Xi+rand*(Xk-Xi);
            End_If
        End_For
        If new learner is better than existing
            Xi=Xnewi;
        End_If
        Xteacher=best learner;
        Xmean=average of learners;
End_While
```

3-3- Routing

We use the TS algorithm for routing and transferring the data collected by the cluster heads to the sink. The TS algorithm consists of a solution that includes a string of position and cost variables. The number of position variables equals the number of cluster heads minus 1 (N$_{Ch}$-1). The figure below shows a view of the solution in the TS algorithm.

Solution	Position	Node 01	Node 02	Node 03	….	Node (n-1)	Node (n)
	Cost						

In the TS algorithm, a number of actions are performed on the solution variables so as to optimize the solution cost. These actions are reversion, swap, and insertion.

To optimize routing using the TS algorithm, we use the Prüfer algorithm [16] to create a tree between the cluster head nodes. This algorithm maps a sequence of numbers to the corresponding tree.

First, we create a solution that assigns a random number between 0 and 1 to each position variable. Then, the solution cost is computed. To calculate the cost of each solution, we first convert it to the corresponding tree using the Prüfer algorithm. Then, the routing is performed according to the obtained tree, and the cost is calculated from (14). E$_1$ is the network energy before applying the routing, and E$_2$ is the computed energy after applying the routing.

$$Cost = E_1 - E_2 \tag{14}$$

Given the actions considered in the TS algorithm, all the states relating to these actions are created in a list named Action List. We perform these actions on the obtained solution and update the cost and position for each action. If a lower cost results, it replaces the best solution, and the corresponding action is placed in the Tabu List and is not performed for a specific number of rounds. The desired number of actions is computed using (15).

$$N_{Action} = N_{Swap} + N_{Reversion} + N_{Insertion}$$

$$N_{Swap} = n \times \frac{(n-1)}{2}$$

$$N_{Reversion} = n \times \frac{(n-1)}{2} \qquad (15)$$

$$N_{Insertion} = n \times n$$

(n is the number of position variables.)

$$Tabu\ List = Round(0.5 \times N_{Action})$$

This is continued until the best solution is obtained. Finally, the obtained solution is given to the Prüfer algorithm, the output of which is an optimal tree according to which the routing is performed. For example, assume the number of cluster heads is 10 in a known round. First, the number of variables inside the solution of the TS algorithm is equal to 9. We consider a random number between 0 and 1 for each variable and compute the initial solution cost.

		01	02	03	04	05	06	07	08	09
Solution	Position	0.37	0.26	0.88	0.76	0.44	0.55	0.45	0.35	0.25
	Cost= 2									

For the cost, we first convert this position into integers and obtain the corresponding array 5 4 10 9 5 7 6 5 4. This array is given to the Prüfer algorithm, and an equivalent tree is created. The cost is equal to the energy consumption of the cluster heads during the transmission of information to the sink according to this tree and route, and we seek to reduce the cost of the problem solution. After the initial solution is known, the desired actions are applied according to the obtained solution and the position and cost of the optimal solution are obtained. if a lower cost results, it replaces the best solution, and the action is not performed for a specific amount of time. This is continued until the cost is optimized. Finally, the obtained solution is converted to a tree via the Prüfer algorithm, which represents our optimal route.

TLSIA Routing Algorithm

```
Create initiate solution;
Sbest=best solution;
While (stopping condition is not met)
    Generate candidate solutions in the neighborhood of Sbest
    For i=candidate solutions
        If candidate_i is not in TabuList
            If candidate_i is better than bestnewsol
                Bestnewsol=candidate_i
            End_If
        End_If
    End_For
    If bestnewsol is better than Sbest
        Sbest= bestnewsol
    End_If
    Push the bestnewsol to TabuList
    If TabuListSize>maxTabuListSize
        Remove the first element from TabuList;
    End_If
End_While
```

3-4- Network Operations and Energy Consumption Computation

The network operations in the proposed algorithm are divided into start-up and register phases. The energy consumption of every node in each round is computed by examining what has occurred in both phases.

3-4-1- Start-up Phase

The sink uses the k_{CP} control packet to communicate with the sensor nodes. These k_{CP} control packets contain short messages that request the ID, position, and the level of energy from each of the sensor nodes. The energy $E_{Rx}(k_{CP})$ is consumed in the process of receiving the control packets from the sink according to (16). Moreover, all the nodes utilize the energy $E_{Tx}(k_{CP}, d)$ to transfer to the sink the control packets that contain data relating to the IDs, positions, and levels of energy.

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

$$E_{Tx}(k,d) = \begin{cases} kE_{elect} + \varepsilon_{mp} k d^4, & if\ d > d_0 \\ kE_{elect} + \varepsilon_{fs} k d^2, & if\ d \le d_0 \end{cases} \qquad (17)$$

Where $d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$ is the threshold distance. The amplifier energy ε_{mp} or ε_{fs} is based on the distance of the receiver and the acceptable bit error. The sink processes the control packets and, according to the proposed algorithm, determines which nodes will be cluster heads and which cluster head each node will become a member of. Moreover, all the nodes (CH or other nodes) use the energy $E_{Rx}(k_{CP})$ to receive their status information from the sink. The energy consumed by the CHs to send TDMA (Time-division multiple access) schedules to their respective members is obtained by the following relationship:

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

The member consumes energy to receive the TDMA schedules from the cluster head, which is computed from (16).

3-4-2- Register Phase

In the register phase, the active nodes send k-bit data to their respective cluster heads in terms of the TDMA schedule they have received from the sink. The cluster head is always ready to receive these sensed data from its members and processes and aggregates all the data received from its members before sending them to the sink. The energy consumed by the cluster head sensor transmitter to perform work, i.e., E_{DA}, is computed from (19).

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

The energy lost in the transmission of the sensed data to the cluster head is calculated using the following relationship:

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

where m_i denotes the member nodes of the series $i = 1, 2, 3, \dots, n - L$, and n and L represent the total numbers of sensor nodes and cluster heads, respectively. The energy consumed by the cluster head to collect the sensed data from the members and itself is determined via (19), as follows.

4- Findings:

All the experiments were conducted within MATLAB R2019b. To prove the efficiency, we compare the proposed algorithm to known protocols such as LEACH, ASLPR, and COARP based on FND, HND, LND, and the total number of data packets received at the sink from the start of the simulation to the end of the network lifetime.

4-1- Network Model Assumptions

The important assumptions for the network model and the radio model in the proposed algorithm are as follows:

- ❖ The sink is a fixed device and a rich source located outside the simulation environment.
- ❖ All the sensors are stable after deployment, and the average energy in the homogeneous or heterogeneous environment is constant.
- ❖ All the sensors are equipped with the Global Positioning System (GPS) or connected to other geographical positioning systems.
- ❖ The communication channel is considered 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
ε_{fs}	10 (pj/bit/m^2)
ε_{mp}	0.0013 (pj/bit/m^4)
E_{elec}	50 (nJ/bit)
E_{da}	5 (nJ/bit)
Data packet size	4100 (bit)

4-2- Simulation Results

In this section, the authors take into account eight scenarios according to Table (4) to evaluate the proposed algorithm. The number of sensors, the size of the environment, and the sink location are the parameters investigated in these scenarios to evaluate the algorithms in which the parameters change in each scenario.

Table 4: Used scenarios

Number	Number of sensors	Network size	Sink location
1	100	200m × 200m	(100m, 250m)
2	100	500m × 500m	(250m, 550m)
3	200	200m × 200m	(100m, 250m)
4	200	500m × 500m	(250m, 550m)
5	500	200m × 200m	(100m, 250m)
6	500	500m × 500m	(250m, 550m)
7	2000	200m × 200m	(100m, 250m)
8	2000	500m × 500m	(250m, 550m)

According to Table (4), the scenarios are simulated in two environments of sizes 200m×200m, and 500m×500m and the number of sensor nodes 100, 200, 500, and 2000, and their results are analyzed. Three factors are investigated in these scenarios: 1) the number of live nodes, 2) energy consumption of the network, 3) packets sent to the sink in each round.

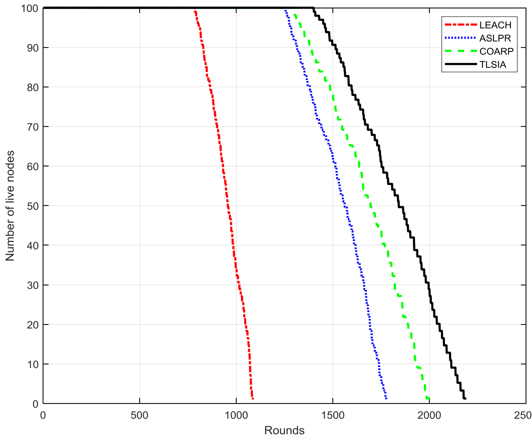


Fig. 4. Number of live nodes in each round in the first scenario.

According to the results obtained in Figure (4) in the first scenario, FND [1], HND [2] and LND [3] in the proposed algorithm are better compared to other approaches and indicates that in the Proposed algorithm, the energy consumption of sensors in each round is less than other methods. In Figure (5), the network's lifetime has been compared; in the Proposed algorithm, the networks' lifetime has increased compared to other methods, which shows the proper performance of the proposed algorithm in clustering and data transmission.

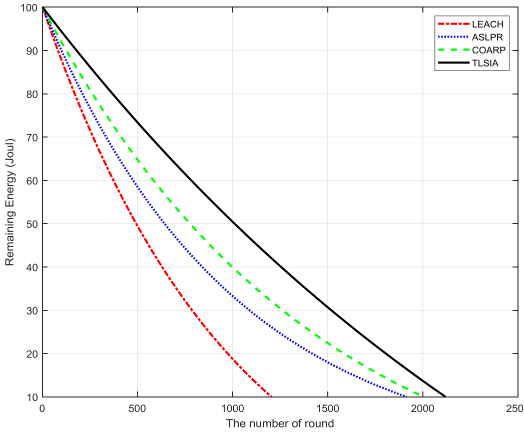


Fig. 5. Network's energy consumption in each round in the first scenario.

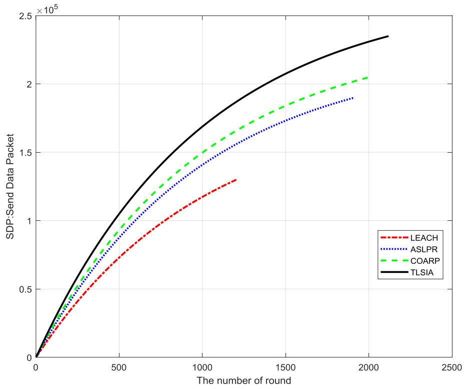


Fig. 6. Packets sent to the sink in each round in the first scenario.

In the simulations, the higher the number of intact packets sent to the sink, the better the performance of the sensor nodes and cluster heads, which leads to an increase in the performance of the wireless sensor network. As shown in Figure (6), in the Proposed algorithm, the number of packets sent to the sink in each round is more than other methods, which indicates the proper performance of the sensor nodes and cluster heads within the wireless sensor network in the TLSIA method.

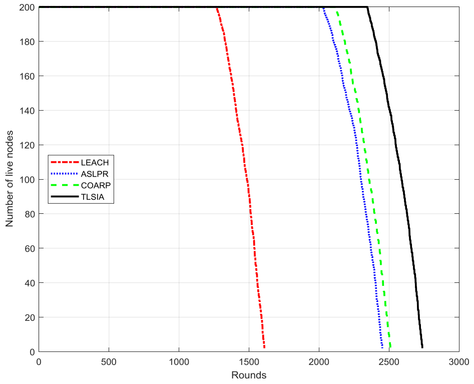


Fig. 7. Number of live nodes in each round in the third scenario.

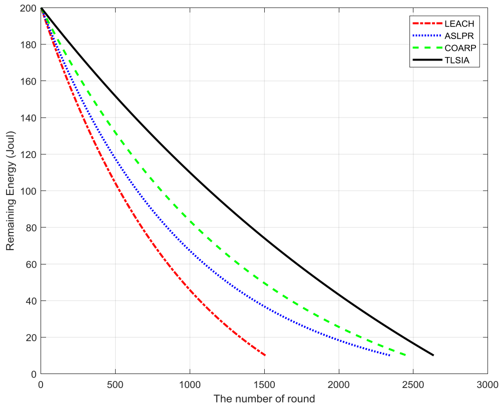


Fig. 8. Network's energy consumption in each round in the third scenario.

[1] First Node Dead
[2] Half Node Dead
[3] Last Node Dead

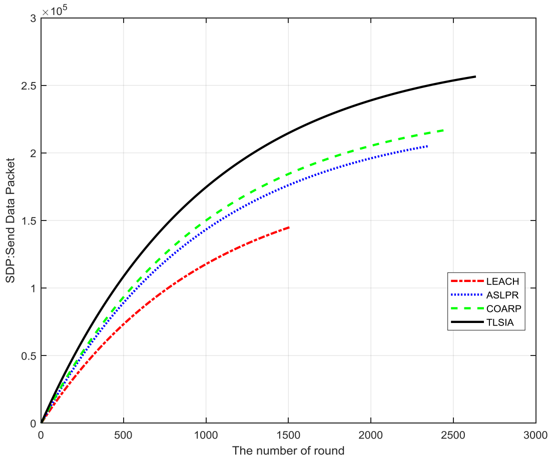


Fig. 9. Packets sent to the sink in each round in the third scenario.

The difference between the first and third scenarios is the number of nodes distributed in the simulation environment. The increase in the number of sensor nodes and the constant size of the simulation environment has led to an increase in the two factors of live nodes and packets sent to the sink in each round, which is true for all comparable algorithms. The results obtained from Figures 7, 8, and 9 indicate that the TLSIA algorithm outperforms the investigated algorithms. This performance includes the number of live nodes, the network's lifetime, and the number of packets sent to the sink in each round.

In Table 5, the authors compare and evaluate FND, HND, and LND factors of the proposed algorithm (TLSIA) compared to other methods in the first four scenarios.

Table 5: Comparison of FND, HND, and LND of TLSIA method with other methods in the first four scenarios.

			Network Size= 200 m × 200 m Sink location= (100 m, 250 m)			Network Size= 500 m × 500 m Sink location= (250 m, 500 m)		
			FND	**HND**	**LND**	**FND**	**HND**	**LND**
Number of sensor nodes	100	**LEACH**	780	1155	1204	2	27	43
		ASLPR	1248	1848	1914	4	43	69
		COARP	1294	1940	2004	5	47	73
		TLSIA	1388	2032	2119	8	54	85
	200	**LEACH**	1507	1431	1268	4	42	61
		ASLPR	2347	2289	2028	7	68	98
		COARP	2456	2361	2117	8	72	105
		TLSIA	2637	2547	2345	11	84	120

The difference between the first, second, third, and fourth scenarios is in the number of sensor nodes and the size of the simulation environment. Increasing the number of network's sensor nodes in these scenarios leads to an increase in the network's lifetime and the number of packets sent to the sink in each round, but increasing the size of the environment leads to a decrease in the network's lifetime and the number of packets sent to the sink in each round. As indicated in Table (5), with increasing the number of sensors as well as the network's size, the TLSIA method is better in terms of FND, HND, and LND in comparison with other techniques. These results mean that the TLSIA algorithm performs better in selecting cluster heads and routing the collected data compared to other methods, which reduces the energy consumption of sensor nodes and cluster heads.

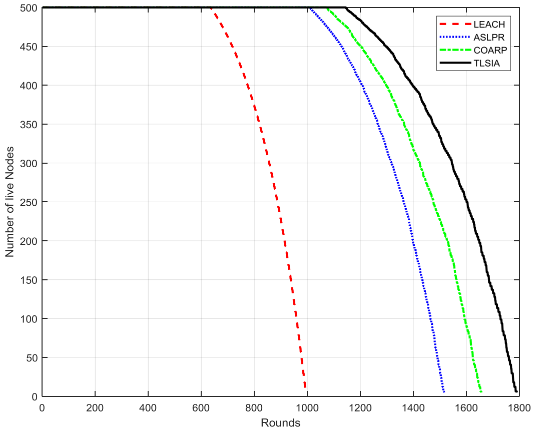


Fig. 10. Number of live nodes in each round in the fifth scenario.

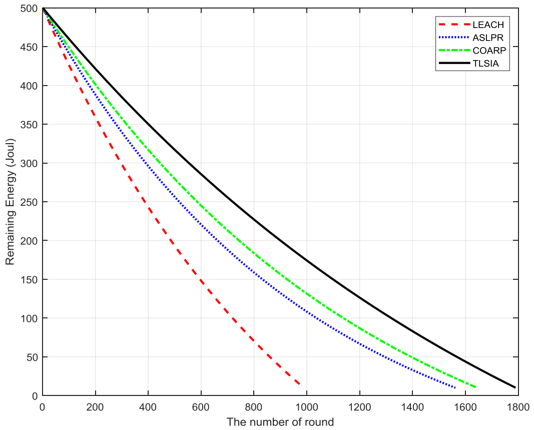


Fig. 11. Network's energy consumption in each round in the fifth scenario.

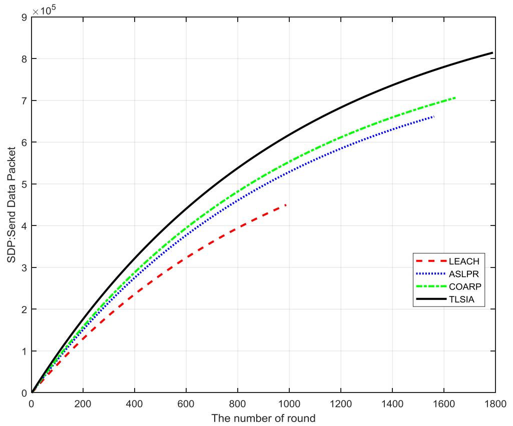


Fig. 12. Packets sent to the sink in each round in the fifth scenario.

In Scenario 5, the number of sensor nodes was considered to be 500, the size of the simulation environment to be 200m×200m, and the nodes being randomly distributed in the environment. According to the results obtained in Figures 10, 11, and 12, it can be concluded that an excessive increase in the number of sensors in the simulation environment has an adverse effect on network's performance since a large number of sensor nodes are distributed in a small environment which leads to an increase in the useless interactions between the sensors as well as an increase in the cluster heads' load, causing energy consumption and rapid discharge of cluster heads. Thus, there must be a tradeoff between selecting the number of sensor nodes and the size of the simulation environment to reach an optimal performance of this network.

In Scenario 7, by increasing the number of sensor nodes to 2000, the results indicate that the TLSIA method is more efficient than other approaches. According to Figures (13), (14), and (15), the TLSIA method outperforms other techniques in terms of the number of live nodes, network's lifetime, and the number of packets sent to the sink.

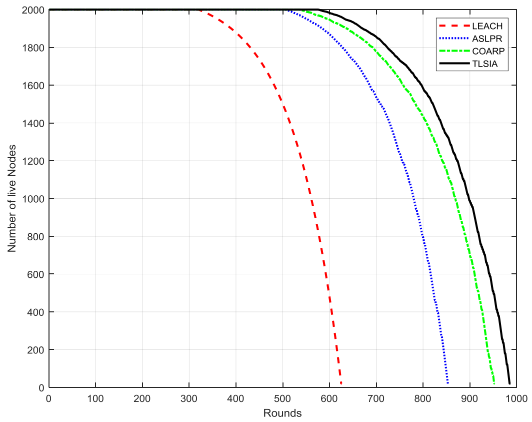


Fig. 13. Number of live nodes in each round in the seventh scenario.

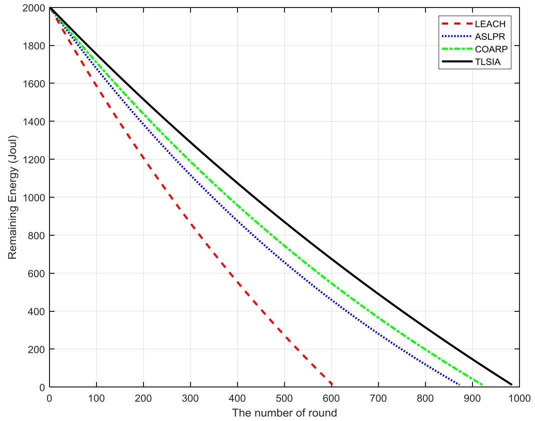


Fig. 14. Network's energy consumption in each round in the seventh scenario.

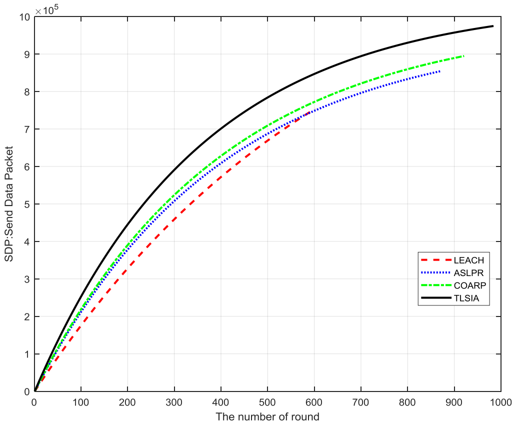


Fig. 15. Packets sent to the sink in each round in the seventh scenario.

In Table (6), the results of the proposed algorithm (TLSIA) are evaluated and compared to other methods in FND, HND, and LND modes. These results are related to the last four scenarios presented in Table (4).

Table 6: Comparison of FND, HND, and LND of TLSIA method with other methods in the second four scenarios.

			Network Size= 200 m × 200 m Sink location= (100 m, 250 m)			Network Size= 500 m × 500 m Sink location= (250 m, 500 m)		
			FND	**HND**	**LND**	**FND**	**HND**	**LND**
Number of sensor nodes	500	**LEACH**	636	910	988	60	102	127
		ASLPR	1005	1456	1562	95	164	205
		COARP	1072	1530	1645	100	174	212
		TLSIA	1145	1657	1790	112	185	234
	2000	**LEACH**	320	405	552	20	30	46
		ASLPR	505	645	872	32	49	74
		COARP	535	680	921	35	52	78
		TLSIA	575	720	984	48	64	92

The results of these four scenarios also indicate that by increasing the number of sensor nodes and also increasing the size of the environment, the TLSIA method has performed better compared to the other methods in terms of the three studied factors: FND, HND, and LND.

By evaluating the proposed scenarios, it can be concluded that some variables such as the size of the simulation environment and the number of sensor nodes distributed in the environment have a significant impact on the energy consumption of the sensors, cluster heads, and the performance of the wireless sensor network. Therefore, one of the significant challenges in such networks is establishing a proper fit between the network size and the number of sensors.

5- Discussion and Conclusion

The wireless sensor networks include a set of sensor nodes designed and applied for particular purposes; hence, energy-saving is considerably important due to the non-rechargeability of sensor nodes. Selecting the appropriate cluster head from the sensor nodes and the hierarchical routing has a significant effect on reducing the energy consumption of the sensor nodes. Different routing protocols, including the LEACH, ASLPR, and COARP protocols, have been proposed to achieve energy efficiency in wireless sensor networks. The purpose of all protocols is to extend the lifetime of wireless sensor networks. In order to achieve this objective, the authors applied the teaching-learning-based optimization algorithm, which consists of two phases: Teaching Phase, Learner Phase. The authors selected the appropriate nodes from the sensor nodes in the network using the TLBO swarm intelligence algorithm, which led to the formation of suitable clusters to reduce the energy consumption of the sensor nodes. After selecting the cluster head and also the clustering operations, the collected data was sent to the sink through a multistage (hierarchical) method by the TS algorithm. This method reduced the energy consumption of the cluster heads when sending data to the sink.

According to the simulation results of the proposed algorithm in this article, the TLSIA algorithm outperformed other compared algorithms in different conditions with increasing the number of sensor nodes and also the size of the simulation environment and also the network's lifetime has been increased. as well as, the proposed TLSIA algorithm can decrease the energy consumption of the nodes and increase the network life. In terms of HND, FND, and LND, the proposed algorithm has had an increase of about 75%, 15%, and 10% compared to the LEACH, ASLPR, and COARP algorithms, respectively. Moreover, the number of packets transmitted to the sink in the proposed algorithm has increased compared to that in other methods. The following suggestions can be made for future works to improve and develop the Proposed algorithm:

1) Mobilization of the sensor nodes inside the network to suitable clustering.
2) Movement of the sink around the network environment to collect the information sensed by the sensor nodes.
3) Use of ensemble learning algorithms in the selection of the cluster head.

6- References

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

[2] 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.

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

[4] Z. Ullah, “A Survey on Hybrid, Energy Efficient and Distributed (HEED) Based Energy Efficient Clustering Protocols for Wireless Sensor Networks,” Wirel. Pers. Commun., vol. 112, no. 4, pp. 2685–2713, 2020.

[5] 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.

[6] S. A. Susan T and B. Nithya, “Cluster Based Key Management Schemes in Wireless Sensor Networks: A Survey,” Procedia Comput. Sci., vol. 171, pp. 2684–2693, 2020.

[7] P. Sarzaeim, O. Bozorg-Haddad, and X. Chu, “Teaching-Learning-Based Optimization (TLBO) Algorithm BT - Advanced Optimization by Nature-Inspired Algorithms,” O. Bozorg-Haddad, Ed. Singapore: Springer Singapore, 2018, pp. 51–58.

[8] M. Gendreau, “An Introduction to Tabu Search,” in Handbook of Metaheuristics, 2006.

[9] U. E. Zachariah and L. Kuppusamy, “A hybrid approach to energy efficient clustering and routing in wireless sensor networks,” Evol. Intell., 2021.

[10] F. Fanian and M. K. Rafsanjani, “Cluster-based routing protocols in wireless sensor networks: A survey based on methodology,” J. Netw. Comput. Appl., vol. 142, pp. 111–142, 2019.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. p. 10 pp. vol.2, 2000.

[12] 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.

[13] 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.

[14] P. K. Roy, C. Paul, and S. Sultana, “Oppositional teaching learning based optimization approach for combined heat and power dispatch,” Int. J. Electr. Power Energy Syst., 2014.

[15] W. Shao, D. Pi, and Z. Shao, “An extended teaching-learning based optimization algorithm for solving no-wait flow shop scheduling problem,” Appl. Soft Comput. J., 2017.

[16] X. Wang, L. Wang, and Y. Wu, “An Optimal Algorithm for Prufer Codes,” JSEA, vol. 2, pp. 111–115, Jan. 2009.

Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 1-10

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Statistical Analysis and Comparison of the Performance of Meta-Heuristic Methods Based on their Powerfulness and Effectiveness

Mehrdad Rohani[1*], Hassan Farsi[1], Seyed Hamid Zahiri[1]

[1].Department of Electronic and Computer Engineering, University of Birjand, Birjand, Iran

Received: 04 May 2021 / Revised: 24 Jul 2021/ Accepted: 24 Aug 2021
DOI: https://doi.org/10.52547/jist.16067.10.37.49

Abstract

In this paper, the performance of meta-heuristic algorithms is compared using statistical analysis based on new criteria (powerfulness and effectiveness). Due to the large number of meta-heuristic methods reported so far, choosing one of them by researchers has always been challenging. In fact, the user does not know which of these methods are able to solve his complex problem. In this paper, in order to compare the performance of several methods from different categories of meta-heuristic methods new criteria are proposed. In fact, by using these criteria, the user is able to choose an effective method for his problem.

For this reason, statistical analysis is conducted on each of these methods to clarify the application of each of these methods for the users. Also, powerfulness and effectiveness criteria are defined to compare the performance of the meta-heuristic methods to introduce suitable substrate and suitable quantitative parameters for this purpose. The results of these criteria clearly show the ability of each method for different applications and problems.

Keywords: Effectiveness; Meta-heuristic Algorithms; Optimization; Powerfulness; Statistical Analysis.

1- Introduction

Optimization is conducting a process to find the best acceptable answer by considering the limits and requirements of the issue. To solve an optimization issue, there might be different answers for the desired optimal parameter, in which function, namely the goal function, is defined to compare these answers and choosing an optimal answer. An optimization method must be able to extract the optimal answer for this function. The advance of the computer in the last five decades leads to the improvement of the optimization methods. Each of these methods has a different ability to solve an optimization problem. However, due to existing the great number of optimization methods, the most important question that arises is which method is suitable, and provides the best performance for solving the problem. These methods can be classified into three broad categories: Enumeration methods, calculates-based methods, and random methods, which are explained in the following.

Enumeration methods: In each iteration, only one point t belonging to the answer space is examined. This category of methods is simpler than the others in terms of implementation but needs considerable calculations. In these methods, no mechanism exists to decrease the scope of the search space, and the scope of the search space is very large. For instance, dynamic programming is an example of these methods that acts completely unintelligent [1, 2].

Calculates-based methods: In these methods, the set of necessary and sufficient conditions are used that apply to the answer to the problem. These methods usually use the gradients of the goal to search. It is possible that sometimes, due to discontinuity of the goal function. Its gradients cannot be calculable. Therefore, these methods also face challenges [3].

Random methods: One of the uncommon methods to find the optimal answer for a problem is to consider all of the possible answers. In this case, the goal function is calculated for all of the possible answers and at the end, the best answer is selected. In this case, the complete count leads to the exact answer to the problem. Using this method in practical problems, is not possible due to the vast range of possible answers for the problem. Given this issue, the effort is always to present methods that have the ability to decrease the search space. To solve this problem, random search methods such as heuristic methods and meta-heuristic methods are presented. This category of

✉ **Hassan Farsi**
Hfarsi@Birjand.ac.ir

methods is able to present a proper answer and close to the optimal answer in a limited time and unlike enumeration and calculates based methods. Random search methods are mainly based on enumeration methods except that they use additional information to guide the search. These methods are completely general in terms of the application area. They are able to solve very complex problems.

The main problem of the heuristic methods is that they get caught in local optimal points and early converge to these points. Meta-heuristic methods are presented to solve this problem [4]. In fact, meta-heuristic methods belong to the category of methods that have a solution to exit local optimal points. These methods are able to be used in a broad range of problems. Optimization methods are used in various fields [5, 6]. The user chooses one of the methods based on the application.

In this paper, we try to examine a set of methods and present statistical analysis. For this reason we determine the stability of each method and their real-time performance. Also, two new criteria powerfulness and effectiveness have been introduced and the performance of each methods has been examined by these two new criteria. In the next section, meta-heuristic methods from different categories are introduced. In the third section, benchmarks as well as conventional evaluation criteria (Best fitness, average run time and standard deviation) are introduced. Also, new criteria and statistical analysis have been introduced in this section. In the fourth section, the tests and results are reviewed.

2- Meta-Heuristic Methods

During the last three decades, the introduction of new meta-heuristic methods and also their application in different devices has been considerably increasing. In 1983, Kirkpatrick proposed the simulated annealing method [7]. In 1992, Koza introduced the genetic programming method [8]. After that Walker et al introduced the first algorithm based on bee colonies in 1993 [9]. In 1994, the term of meta-heuristic was used by Golver when introducing the tabu search method [10]. In 1995, Kennedy and Eberhat introduced particle swarm optimization [11]. In 2002, Passino introduced the bacterial foraging-based method [12]. In 2008, a bio-geography-based optimization algorithm was introduced by Simon [13]. Many methods in this area have been presented in the last decade. In 2014, the grey wolf algorithm was introduced by Mirjalili et al. [14]. In 2015, the Ant Lion Optimizer (ALO) was presented by Mirjalili [15]. In the same year, the moth-flame optimization algorithm was presented by Mirjalili [14]. In 2016, Mirjalili introduced dragonfly (DA), multi-verse (MVO), sine cosine (SCA), and whale optimization (WOA) algorithms for optimization [17-20]. In 2017, the salp swarm algorithm (SSA) was presented by Mirjalili [21]. Development of these algorithms has occurred at a high speed in recent years.

These methods are able to exit the local optimal answers and move to the global optimal answer in a short period of time. The important factor in these methods is the dynamic balance between the exploration and exploitation strategies. Exploration is able to properly search the answer space. Exploitation strategy performs the search operation in spaces with more possibility and prevents the loss of time in search space in which the possibility of the optimal answer is low. Meta-heuristic methods are divided by categories of methods including methods based on single-point and based on population, inspired by nature and without inspiration by nature, with memory and memoryless, and probabilistic-definitive. Some of the meta-heuristic methods are memoryless. They do not use the information obtained during the search. Some of them take advantage of the obtained information. Single point-based methods change an answer during the search process, while population-based methods consider a set of answers during the search [22]. Generally, this type of method is slower comparing to the single-point methods but they are able to produce more desirable answers. However, due to the advance in computer calculation power, the population-based methods hold more importance. Among these methods, WOA, GWO, DA and SSA can be pointed out. All of these methods belong to the category of nature-inspired methods which have memory and they are also based on population. Another category of these methods is presented in figure 1 [23].

2-1- Whale Optimization Algorithm (WOA)

This method is inspired by social behavior, the mechanism of which is based on the social movement of humpback whales and how they hunt. Humpback whales are able to identify the location of the prey [20]. The primary location of the search agents in modeling the algorithm is considered to be close to the desired situation, and after determining the best search agent, other agents update their location according to that. This behavior is modeled through the Eq. (1) and Eq. (2). All the equations in this subsection are adopted from [20].

$$\vec{D} = \left|\vec{C}\vec{X}(t) - \vec{X}(t)\right| \quad (1)$$

$$\vec{X}(t+1) = \overrightarrow{X^{*}}(t) - \vec{A}.\vec{D} \quad (2)$$

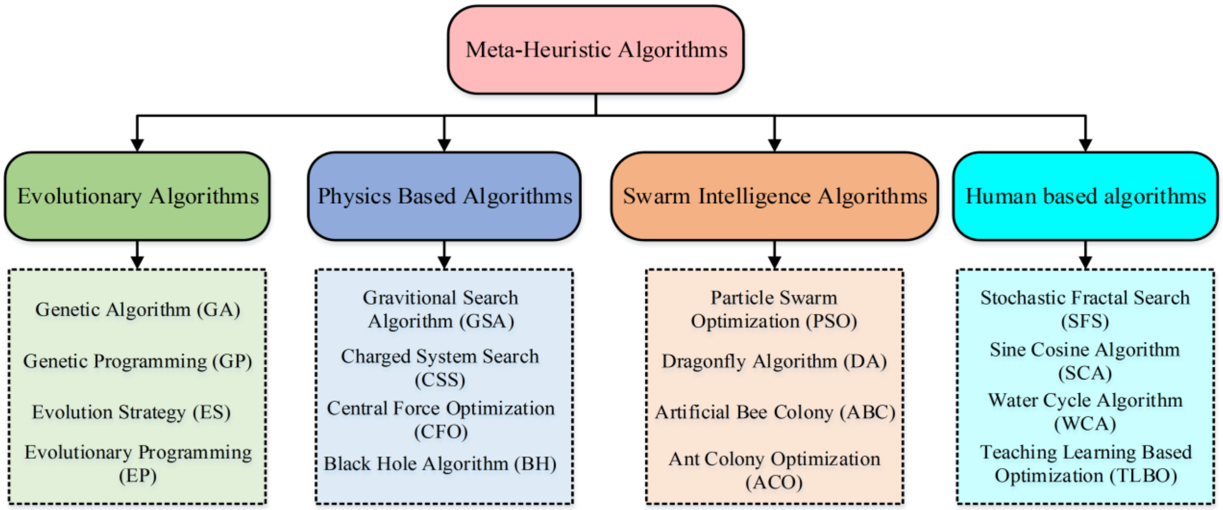

Meta-Heuristic Algorithms
Evolutionary Algorithms
Physics Based Algorithms
Swarm Intelligence Algorithms
Human based algorithms
Genetic Algorithm (GA)
Genetic Programming (GP)
Evolution Strategy (ES)
Evolutionary Programming (EP)
Gravitional Search Algorithm (GSA)
Charged System Search (CSS)
Central Force Optimization (CFO)
Black Hole Algorithm (BH)
Particle Swarm Optimization (PSO)
Dragonfly Algorithm (DA)
Artificial Bee Colony (ABC)
Ant Colony Optimization (ACO)
Stochastic Fractal Search (SFS)
Sine Cosine Algorithm (SCA)
Water Cycle Algorithm (WCA)
Teaching Learning Based Optimization (TLBO)

Fig. 1. Classification of meta-heuristic algorithms [23]

Where t is the current iteration, $\vec{A}$ and $\vec{C}$ are the coefficient vectors, $\overrightarrow{X^*}$ is the location vectors for the best response obtained in the t iteration and $\vec{X}$ is the location vector. Also, $\overrightarrow{X^*}$ must be updated in each iteration in case of the existence of a better answer. Coefficient vectors $\vec{A}$ and $\vec{C}$ are calculated with Eq. (3) and Eq. (4).

$$\vec{A} = 2\vec{a}.\vec{r}.\vec{a} \tag{3}$$

$$\vec{C} = 2\vec{r} \tag{4}$$

Where a is decreased linearly from 2 to zero with the increase in iterations. Also, the vector r is a random number in the range between 0 and 1.

Bubble-net attacking method (exploitation phase): In this attack, two methods of contraction blocking mechanism according to Eq. (1) and spiral updating location according to Eq. (2) are used. There is a 50% probability that the whale will choose one of these two mechanisms, and there is a 50% possibility that the whale chooses one of these two mechanisms. Each of these behaviors is mathematically modeled by Eq. (5) and Eq. (6).

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A}.\vec{D} \qquad if\ p \geq 0.5 \tag{5}$$

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) \quad if\ p < 0.5 \tag{6}$$

Where p is a random number between zero and one, b is a constant to define a logarithmic spiral shape, and it is a random number between 1 and -1, $\overrightarrow{D'}$ is the distance of the lth whale to the prey, which is represented as Eq. (7).

$$\overrightarrow{D'} = \overrightarrow{X^*}(t) - \vec{X}(t) \tag{7}$$

Search for prey (exploitation phase): In this method, random selection is used to update the search agents to let the algorithm perform a global search in the search space, which is modeled by Eq. (8) and Eq. (9).

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \tag{8}$$

$$\vec{D} = |\vec{C}.\overrightarrow{X_{rand}} - \vec{X}| \tag{9}$$

Where X_{rand} is the selected random location between the current population and $|A| \geq 1$ is chosen to make the search agents perform a global search.

2-2- Grey Wolf Optimization Algorithm (GWO)

Footnotes should be typed in singled-line spacing at the bottom of the page and column where it is cited. Footnotes should be rare. The grey wolf method is one of the methods inspired by social behavior. Grey wolf is considered as the highest-ranking hunter since there is no natural hunter for these animals. These wolves live in a group of 5 or 15, and their leader is recognized as alpha wolf (α). Alpha wolves are not the strongest members of the group but they are the best at managing it. The second category in terms of the hierarchy is the beta (β), which helps α in decision makings. If it is necessary, the β wolves who replace the alpha. The third category is the delta wolves (δ), and after that, is the omega (ω) wolves. To model the social behavior of the wolves, a random population is generated. The grey wolf algorithm takes advantage of three answers, namely alpha, beta, and delta, and omega answers follow these three answers [14]. To model the three phases, first, it is necessary for the points around the prey to be predetermined and then the wolves will start moving toward it and at the end, the attack will start. Location vectors of the wolves are modeled using the Eq. (10) and Eq. (11). All the equations in this subsection are adopted from [14].

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \tag{10}$$

$$\vec{D} = |\vec{C}.\overrightarrow{X_{rand}} - \vec{X}| \tag{11}$$

Where t is the running iteration, $\vec{A}$ and $\vec{C}$ are the coefficient vectors, $\overrightarrow{X_P}$ is the prey vector, and $\vec{X}$ indicates the location vector of a grey wolf. Vectors A and C are calculated using the Eq. (12) and Eq. (13).

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1}.\vec{a} \quad (12)$$

$$\vec{C} = 2\overrightarrow{r_2} \quad (13)$$

Where a reduces linearly from 2 to 0 with the increase of the iterations. Also, r_1 and r_2 vectors are random numbers between 0 and 1. To mathematically model the behavior of the grey wolves, it is assumed that the alpha (the best candidate), beta, and delta have the best information about the prey. Therefore, the three obtained answers are saved and the location of the other wolves is updated based on these answers. The Eq. (14) and Eq. (15) indicate the modeling of the wolves' behavior.

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1}\overrightarrow{X_\alpha}(t) - \vec{X}|, \overrightarrow{D_\beta} = |\overrightarrow{C_2}\overrightarrow{X_\beta}(t) - \vec{X}|, \overrightarrow{D_\delta} = |\overrightarrow{C_3}\overrightarrow{X_\delta}(t) - \vec{X}| \quad (14)$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}, \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}, \overrightarrow{X_3} = \overrightarrow{X_\beta} - \overrightarrow{A_3} \quad (15)$$

After calculating theX_i, point X will be updated as Eq. (16) in the next level.

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \quad (16)$$

Finally, after finishing the iterations, the location of the alpha wolf will be selected as the optimum point.

2-3- Dragonfly Optimization Algorithm (DA)

The dragonfly algorithm is based on mass intelligence and inspired by nature. The main idea is based on the behavior of dragonflies while hunting for food and prey [17]. The mass behavior and mass formation of the dragonflies are performed for two purposes: prey which is called the static mass or nutrition, and the immigrant or dynamic mass. These behaviors are modeled by the Eq. (17). All the equations in this subsection are adopted from [17].

$$S_i = \sum_{j=1}^{N} X - X_j$$

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N}$$

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \quad (17)$$

Where S_i is separation, A_i is alignment, and C_i is cohesion. X is current location, N is the number of neighbors, X_j is the jth neighbor, and V_j is the speed of the jth neighbor. The location of the food (goal) and the enemy (search agent) is modeled by Eq. (18) and Eq. (19).

$$F_i = X^+ - X$$

$$E_i = X^- + X \quad (18)$$

Where X is the location of the search agent. X^+ is the location of the food source, and X^- is the location of the enemy. The behavior of the dragonflies is modeled by Eq. (19).

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (19)$$

s, a, c, f and e are the weight values for adjusting the exploration and exploitation processes. The parameter W is the weight of inertia and t is the iteration counter. The position of each search agent is expressed according to the Eq. (20).

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (20)$$

To improve the exploration in the search space and modeling the static behavior of the dragonflies, when there is no neighbor the random walking process is added to this algorithm to update the location of the dragonflies according to the Eq. (21).

$$X_{t+1} = X_t + Lavy(d) \times X_t \quad ; \quad Lavy(d) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{\frac{1}{\beta}}} \quad (21)$$

Where d is the dimension of the location vector and $Lavy(d)$ forms the random walking process, where r_1 and r_2 are random numbers between 0 and 1, and β is a constant number that is assumed to be 1.5. σ is calculated according to the Eq. (22) where $\Gamma(x) = (x-1)!$.

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \quad (22)$$

Finally, the location of the dragonflies is updated based on the two static and dynamic behaviors and the best answer is selected as the optimal answer to the problem [17].

2-4- Salp Swarm Optimization Algorithm (SSA)

This method is inspired by the social behavior of the salps. These creatures move in the deep waters in groups and under the name of salp chain. This behavior, as some researchers believe, is for better movement and fast access to food [21]. To model the behavior of the salps, they are divided into two groups of leader and followers. The leader is the first member of the salps chain and the others are called the followers. The food source for the salps is known as the F matrix, and the location of the salps is modeled by the Eq. (23). All the equations in this subsection are adopted from [21].

$$X_j^1 = \begin{cases} F_j + c_1\left((ub_j - lb_j)c_2 + lb_j\right) & c_3 \geq 0 \\ F_j - c_1\left((ub_j - lb_j)c_2 + lb_j\right) & c_3 < 0 \end{cases} \quad (23)$$

Where X_j^1 is the location of the first salp (leader) in the jth dimension, F_j is the location of the food in the j th dimension, ub_j is the upper bound of the jth dimension, and lb_j is the lower bound of the jth dimension. The parameters c_2 and c_3 are random numbers between 0 and 1, but the parameter c_1 has an important role, namely exploration, in the search space, and is modeled by Eq. (24).

$$c_1 = 2\mathrm{e}^{-(\frac{4l}{L})^2} \quad (24)$$

Where l is the running iteration and L is the maximum iteration. The location of the follower salps is expressed by Eq. (25).

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \quad (25)$$

Where $i \geq 2$ and x_j^i is the location of the ith follower salp in the jth dimension. In this way, the SSA algorithm updates the location of the leader and its followers in each iteration and introduces the best location as the best answer in the last iteration [21].

3- Stability Analysis of Meta-Heuristic Methods Based on Statistical Methods

So far, different methods have been reported to evaluate the stability of meta-heuristic methods based on statistical analysis. In this section, we introduce the proposed method while referring to the methods that have been expressed so far in research.

3-1- Conventional Stability Analysis Methods

Stability analysis of the meta-heuristic methods is performed using two methods of mathematical analysis and statistical analysis. Dorigo in the optimization method of ants ACO proved by limiting the pheromone that the ant method converges to the optimal answer, and the method will be stable [24]. Clerc performed statistical analysis for the particle swarm method and guaranteed the convergence under some conditions for the available parameters in the problem [25]. In this research, Monte Carlo evaluation is used to analyze the stability of the meta-heuristic methods. This analysis includes the examination of convergence in different optimization problems. Also, the ability of each method in encountering each of these problems is determined with high confidence through many iterations. The time parameter is also calculated for each algorithm so that a user can have the ability to choose a method based on online and offline applications. Also, two new metrics namely powerfulness and effectiveness are introduced to examine the convergence and power of a method in obtaining the global answer.

3-1-1- Benchmark Functions

Researchers face varies problems with different complexities. They use different meta-heuristic algorithms to solve and optimize their problems. We have used 23 benchmark functions with different complexities including unimodal, multimodal, and fixed-dimension. The performance of each algorithm has been tested by solving these 23 problems [26-28]. These functions are presented in table 1-3. Fmin indicates the optimum point in the search space. Table 1 indicates the F1-F7 functions that have one global optimum point, and search the exploitation process in the search space, and test the performance of a method. Benchmark functions of F8-F13 are multimodal functions and they are shown in table 2. This category has several local optimum points, and these local optimum points exponentially increase with respect to the increase in the dimension. These functions are perfectly capable of test the exploration process in the search space. The third category is the fixed-dimension multimodal. These functions also have one global optimum point and several local optimum points that, similar to the second category, analyze the effectiveness and powerfulness of a method. These are shown in table 3. Figure 2 indicates the 3-D representation of the benchmark functions' search space.

Table 1: Description of unimodal benchmark

Function	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[-100,100]	0
$F_4(x) = max_i\{\lvert x_i \rvert, 1 \leq i \leq n\}$	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	[-30,30]	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	[-100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	[-1.28,1.28]	0

Although these answers have been reported in the base articles, since meta-heuristic algorithms act randomly, a method has to be repeated a considerable number of times and an average of the result has to be presented. As such, the average of the best answer has been calculated for each algorithm with 1000 iterations for each run and 100 runs for each algorithm. These results are reported in table 4, table 5, table 6 and table 7. Also the convergence curves are reported in red color for each method on benchmark functions in figure 3.

3-1-2- Best Fitness Metric

The first assessment metric of every method is the best answer of the algorithm to each benchmark function.

3-1-3- Average Run Time Metric

The second assessment metric is the run time of each algorithm. In this metric, the average of the run time of each algorithm for convergence to 5% range of the best answer obtained has been calculated and reported in the tables. This metric can affect the selection of a user when using an optimization method, and, depending on the application, it can be a tradeoff between the best answer and the run time of the algorithm.

3-1-4- Standard Deviation Metric

The next analysis is the standard deviation parameter. This parameter indicates the deviation rate of the calculated answers and the reliability of a method.

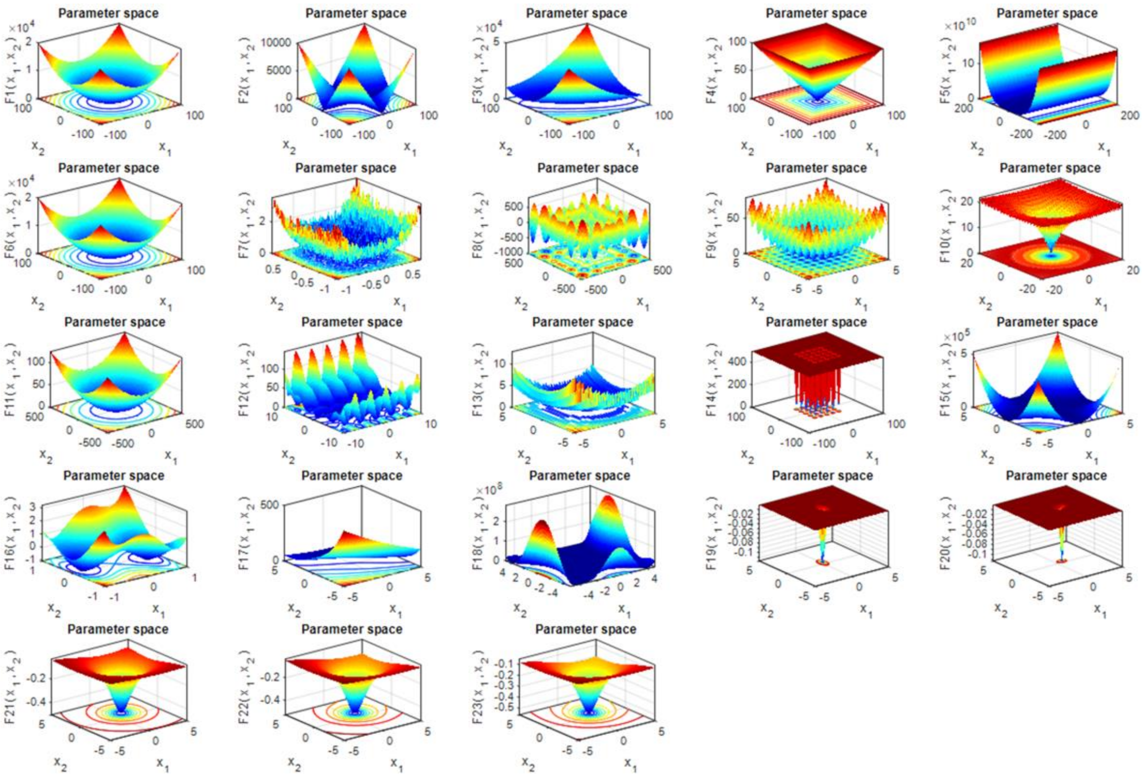


Fig. 2. Search space of composite benchmark functions.

Table 2: Description of multimodal benchmark

Function	Range	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	[-500,500]	-418.98×5
$F_9(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	[-32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]	0
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $k(x_i - a)^m \quad x_i > a$ $0 \quad -a < x_i < a$ $k(-x_i - a)^m \quad x_i < -a$	[-50,50]	0
$F_{13}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	[-50,50]	0

Table 3: Description of fixed-dimension multimodal benchmark

Function	Range	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i + a_{ij})^6}\right)^{-1}$	[-65, 65]	1
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	[-5,5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	[-5,5]	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	[-5,5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[-2,2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	[0,1]	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	[0,1]	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	-10.5363

3-2- The Proposed Stability Analysis Methods

3-2-1- Powerfulness (PF)

A new metric namely the powerfulness is introduced to investigate the convergence of each method. The number of search agents has been considered to be 30 for all methods. Each search agent has fitness in each iteration of the experiment. With increasing the number of iterations and progress of the algorithm, the fitness values are updated in each iteration. In this metric, fitness is investigated after 1000 iterations. If the amount of algorithm fitness is convergence to the global answer with a 0.001 precision, then that algorithm has obtained the desired answer. In Eq. (26), N is equal to 100. This means that the algorithm is repeated 1000 times in each run and this operation will be averaged 100 times. The sign 1(.) will be equal to 1 if the condition inside the parentheses is satisfied. In Eq. (27), f_{global} indicates the global optimum answer and the $f_{calculated}$ shows the calculated optimum answer.

$$PF = Powerfullness = \frac{\sum_{i=1}^{N} 1(\Delta f_i \leq 0.001)}{N} \quad (26)$$

$$\Delta f_i = |f_{global} - f_{calculated}| \quad (27)$$

3-2-2- Effectiveness (EF)

Effectiveness has been introduced to investigate the powerfulness of an algorithm in obtaining the global answer. In this metric, the fitness of all of the search agents is examined after 1000 iterations. If the fitness rate of all of the search agents is exactly equal to the global answer, then that algorithm has obtained the desired answer. In Eq. (28), N is equal to 100. This means that the algorithm is repeated 1000 times in each run and this operation will be averaged 100 times. The sign 1(.) will be equal to 1 if the condition inside the parentheses is satisfied. In Eq. (29) f_{global} indicates the global optimum answer and the $f_{calculated}$ shows the calculated optimum answer.

$$EN = Effectiveness = \frac{\sum_{i=1}^{N} 1(\Delta f_i = 0)}{N} \quad (28)$$

$$\Delta f_i = |f_{global} - f_{calculated}| \quad (29)$$

4- Experimental Results

In this section, the results of the introduced different analysis are presented. The obtained results are presented in tables and graphs. The tables 4-7 also include two sections. The first section of the tables includes the metrics of the best answer, standard deviation, and average time for 23 benchmark functions in different columns. The analysis of each method is reported in a separate table. The column “Reported” is the obtained results of the main reference of each method that has reported the average values of the best answer (Avg Best) and the deviation rate (Avg Std). The column “Calculated” is the results of the performed runs in this research that include the average of the best answer (Avg Best), standard deviation (Avg Std), and average run time (Avg Time). In the second part of the tables including two columns indicates the new metric namely the effectiveness and powerfulness on 23 functions for each algorithm in the “EN” and “PF” columns.

As an instance, three functions F4, F9 and F15 were selected from 23 benchmark functions. In figure 3, convergence curves are shown for 1000 iterations and 100 runs on each algorithm.

4-1- Discussion on Results

In this section, some tables are provided to examine the obtained results and to summarize the large amount of the obtained data. In these tables, we have tried to evaluate the results in a way so that the user can easily select the method in accordance with their needs and application. In this section, the averaging operation is performed on the F1-F7 benchmark functions called unimodal, F8-F13 functions called the Multimodal, and F14-F23 functions named the Fixed-dimension multimodal. The highest value is specified as Bold and the lowest value is specified as Underline in each column.

Average PF criterion

The PF criterion is investigated in Table 8. In this table, the DA method has been able to provide the best result for the unimodal functions with 100 % value. The GWO, and SSA methods are next in the ranking with 85.71%. Also, the WOA method with 59.85% has the lowest amount in the PF criterion. The SSA method in multi-modal functions could have the best performance in this type of function with 100%. In these functions, the WAO, GWO and DA methods have the equal ranking with 83.33%. The best value for the fixed-dimension multimodal functions is 100% for the SSA and DA methods while the WOA method has the lowest amount with 80.70%. As observed from table 8, the WOA method has the weakest performance in the PF metric.

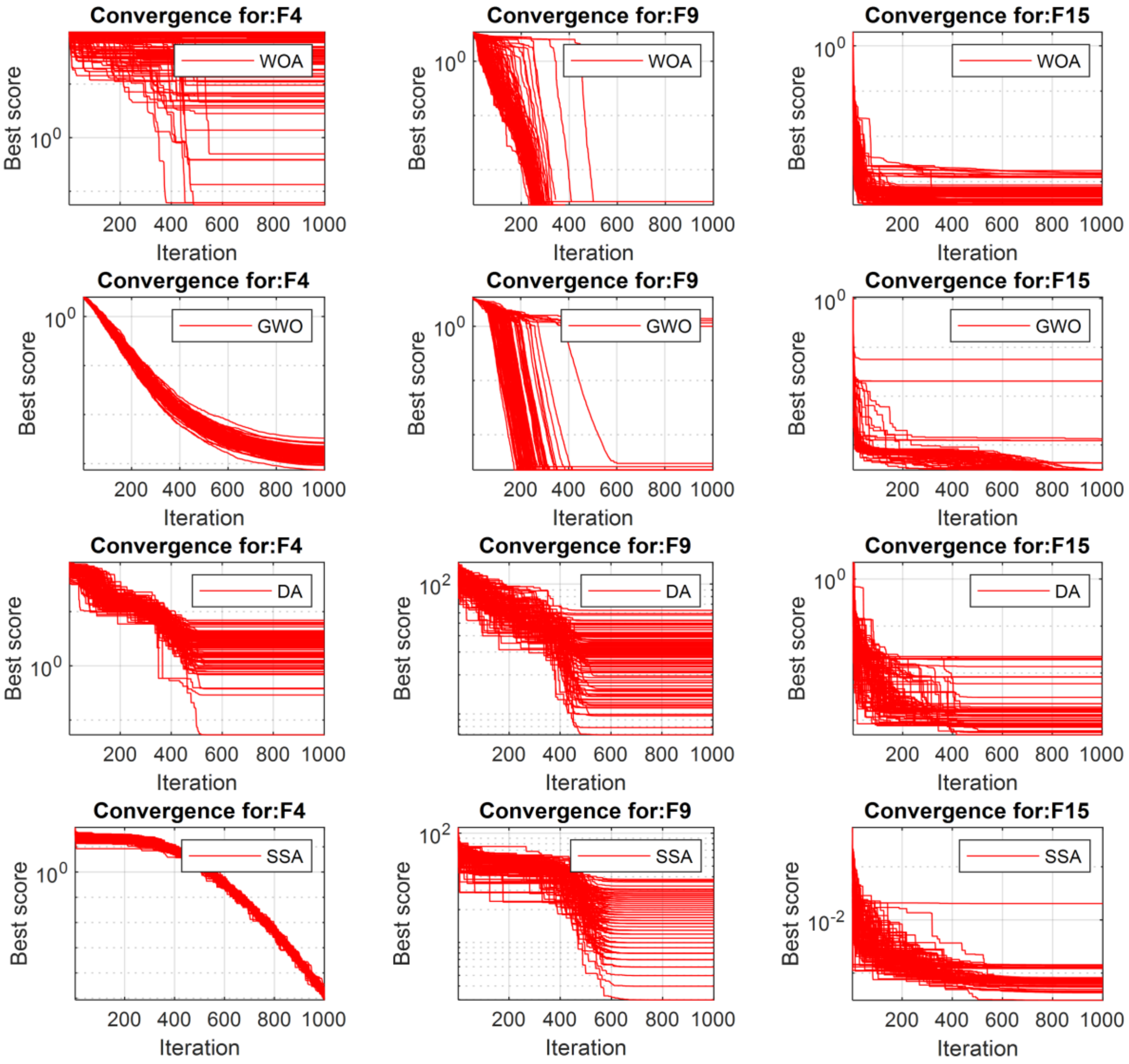


Fig.3. Convergence curves of WOA, GWO, DA and SSA algorithms that obtained from F4, F9 and F15

Table 4: WOA algorithm

WOA	**Reported**		**Calculated**				
Func Test	**Avg Best**	**Avg Std**	**Avg Best**	**Avg Std**	**Avg Time**	**EN**	**PF**
F1	1 .41E −30	4 .91E −30	6.59E-149	4.59E-148	7.70E-02	100	100
F2	1 .06E −21	2 .39E −21	1.17E-102	1.09E-101	8.33E-02	100	100
F3	5 .39E −07	2 .93E −06	20783.72516	10157.99167	0.296841885	0	0
F4	0 .072581	0 .39747	36.54531187	31.21106177	0.048915693	0	19
F5	27 .86558	0 .763626	27.13631191	0.510781163	0.06949779	0	100
F6	3 .116266	0 .532429	0.086487861	0.123716391	0.043686051	0	100
F7	0 .001425	0 .001149	0.001603156	0.001861557	1.42E-01	0	0
F8	−5080.76	695.7968	-10989.67576	1612.087325	0.055816373	0	0
F9	0	0	0	0	0.030713043	100	100
F10	7 .4043	9 .897572	4.09E-15	2.23E-15	8.11E-02	100	100
F11	0 .0 0 0289	0 .001586	0.006625063	0.026231674	0.024578934	94	100
F12	0 .339676	0 .214864	0.00603913	0.010095281	0.297053286	5	100
F13	1 .889015	0 .266088	0.222387626	0.178598496	0.271053991	0	100
F14	2 .111973	2 .498594	2.265421628	2.647242836	0.232368918	0	100
F15	0 .0 0 0572	0 .0 0 0324	0.000663949	0.000350605	2.88E-02	87	100
F16	−1 .03163	4 .2E −07	-1.031628453	5.10E-11	5.08E-02	99	100
F17	0 .397914	2 .7E −05	0.397887754	9.41E-07	3.07E-02	18	100
F18	3	4 .22E −15	3.00004213	0.000223778	1.14E-02	3	100
F19	−3 .85616	0 .002706	-3.859688471	0.003343901	2.80E-02	12	100
F20	−2 .98105	0 .376653	-3.194394261	0.255546232	0.032039408	7	100
F21	−7 .04918	3 .629551	-8.835498564	2.475814202	0.02778591	0	19
F22	−8 .18178	3 .829202	-8.194635405	2.801778303	0.052576967	0	43
F23	−9 .34238	2 .414737	-8.759948348	2.843792657	0.061611228	0	45

Table 5: GWO algorithm

GWO	**Reported**		**Calculated**				
Func Test	**Avg Best**	**Avg Std**	**Avg Best**	**Avg Std**	**Avg Time**	**EN**	**PF**
F1	6.59E-28	6.34E-05	4.28E-59	1.08E-58	2.35E-01	100	100
F2	0.029014	0.042144	1.24E-34	1.86E-34	2.31E-01	100	100
F3	3.29E-06	79.14958	1.19E-14	5.91E-14	3.80E-01	100	100
F4	5.61E-07	1.315088	1.51E-14	2.84E-14	2.13E-01	100	100
F5	26.81258	69.90499	26.69458625	0.742391628	0.216549825	0	100
F6	0.816579	0.000126	0.621146399	0.371011473	0.204170201	6	100
F7	0.002213	0.100286	0.000871956	0.000492435	2.58E-01	0	0
F8	-6123.1	-4087.44	-5928.936636	879.2135455	0.262058383	0	0
F9	0.310521	47.35612	0.405729452	1.451143899	0.04926481	93	100
F10	1.06E-13	0.077835	1.64E-14	2.88E-15	1.21E-01	100	100
F11	0.004485	0.006659	0.003455183	0.008551483	5.97E-02	85	100
F12	0.053438	0.020734	0.038594141	0.02273283	0.520416024	0	100
F13	0.654464	0.004474	0.546402772	0.199821466	0.513642556	2	100
F14	4.042493	4.252799	3.855341713	4.114698941	0.191481026	0	100
F15	0.000337	0.000625	0.004964973	0.008460522	2.60E-02	77	100
F16	-1.03163	-1.03163	-1.031628448	5.88E-09	7.65E-02	100	100
F17	0.397889	0.397887	0.397898355	7.68E-05	5.67E-02	86	99
F18	3.000028	3	3.000009343	1.19E-05	3.46E-02	22	99
F19	-3.86263	-3.86278	-3.861542428	0.002668951	4.39E-02	1	100
F20	-3.28654	-3.25056	-3.26296912	0.093013085	0.05581884	0	100
F21	-10.1514	-9.14015	-9.450634359	1.847937725	0.095443428	0	100
F22	-10.4015	-8.58441	-10.19071187	1.042754866	0.128936701	0	100
F23	-10.5343	-8.55899	-10.15788663	1.53690521	0.138736578	0	100

Table 6: WOA algorithm

DA	**Reported**		**Calculated**				
Func Test	Avg Best	Avg Std	Avg Best	Avg Std	Avg Time	**EN**	**PF**
F1	2.85E-18	7.16E-18	8.709019492	8.470920307	8.907355455	0	100
F2	1.49E-05	3.76E-05	1.541030894	1.180641229	10.18661535	0	100
F3	1.29E-06	2.10E-06	174.129871	375.1554082	7.222378913	1	100
F4	0.000988	0.002776	2.329263546	1.416677941	11.25669589	0	100
F5	7.600558	6.786473	891.914574	1802.132876	9.897716262	1	100
F6	4.17E-16	1.32E-15	8.566349466	13.79450972	11.06746413	0	100
F7	0.010293	0.004691	0.019568582	0.012024944	8.895041751	0	100
F8	-2857.58	383.6466	-2818.040967	355.3376662	15.73808298	0	0
F9	16.01883	9.479113	27.50861574	11.72206506	13.87184983	0	100
F10	0.23103	0.487053	2.6695387	1.019690292	12.87132673	0	100
F11	0.193354	0.073495	0.777684867	0.300217353	11.87462016	3	100
F12	0.031101	0.098349	1.682915888	1.241529733	11.20859426	0	100
F13	0.002197	0.004633	0.618542299	0.541132351	10.60521735	0	100
F14	103.742	91.24364	1.047468983	0.406581789	1.589713017	0	100
F15	193.0171	80.6332	0.003213229	0.005656206	6.15745856	0	100
F16	458.2962	165.3724	-1.031626118	7.82E-06	4.344534415	37	100
F17	596.6629	171.0631	0.397888712	5.26E-06	4.605963481	0	100
F18	229.9515	184.6095	3.000012725	3.43E-05	4.557858122	0	100
F19	679.588	199.4014	-3.862609267	0.000666809	5.160709293	0	100
F20	2.85E-18	7.16E-18	-3.254038178	0.095517834	12.11661363	0	100
F21	1.49E-05	3.76E-05	-7.423171	2.765308816	8.4690914	12	100
F22	1.29E-06	2.10E-06	-8.53677162	2.682979658	7.306447437	15	100
F23	0.000988	0.002776	-8.216681515	2.963678316	8.738156894	24	100

Table 7: GWO algorithm

SSA	**Reported**		**Calculated**				
Func Test	Avg Best	Avg Std	Avg Best	Avg Std	Avg Time	**EN**	**PF**
F1	0	0	1.27E-08	3.54E-09	0.181400388	100	100
F2	0.2272	1	0.009109382	0.066197136	0.102852086	97	100
F3	0	0	2.41E-09	1.54E-09	0.209767524	100	100
F4	0	0.6556	1.55E-05	4.02E-06	0.131577107	100	100
F5	0	0	97.88033843	301.1603942	0.104756503	0	100
F6	0	0	6.69E-10	2.52E-10	0.139161432	100	100
F7	0.0028	0.007	0.005825155	0.003611733	0.141422613	0	0
F8	1	0.0071	-2843.882009	300.9224241	0.108420644	0	100
F9	0.4254	0.9502	17.92912747	7.936964991	0.104917015	0	100
F10	0.0598	0.5279	0.692505073	0.860483282	0.11545747	58	100
F11	0	0	0.201908936	0.084030016	0.152391405	0	100
F12	0	0	0.39478531	0.806546346	0.228802092	72	100
F13	0	0	0.002297837	0.004722304	0.249327454	82	100
F14	0.0557	0.809	1.017884379	0.13986487	0.186448041	0	100
F15	0	0	0.001850349	0.004276931	0.027172082	73	100
F16	0.1952	0.1527	-1.031628453	8.35E-15	0.099933664	0	100
F17	0	0.0651	0.397887358	1.65E-14	0.090015482	0	100
F18	0.1417	0.5571	3	9.31E-14	0.108162701	0	100
F19	0.0832	0.7059	-3.862782148	5.77E-14	0.139162503	0	100
F20	_	_	-3.22247955	0.046978076	0.121644601	0	100
F21	_	_	-8.340784046	2.85966839	0.10572867	70	100
F22	_	_	-8.715288573	2.853571407	0.104817694	80	100
F23	_	_	-9.445250741	2.555722593	0.129785398	83	100

Table 8: Average PF criterion

	Function type		
Algorithm	**Unimodal**	**Multi modal**	**Fixed-dimension multimodal**
WOA	59.85	83.33	80.70
GWO	85.71	83.33	99.80
DA	**100**	83.33	**100**
SSA	85.71	**100**	**100**

Besides the powerfulness metric, the convergence rate of each method on each benchmark function can be examined by convergence graphs depicted using red color in figure 3. This test is performed visually. By observing any red convergence graph it can be realized that how cohesive an algorithm behaves on a benchmark function. If the graphs are drawn coherently and close to each other, the convergence rate is high for example GWO and SSA methods on F4 function. On the contrary, if the drawn lines are at a distance from each other, it shows the different behavior of each algorithm on a benchmark function. This means that an algorithm results in different answers in each run and the reliability on that benchmark function is decreased by that algorithm for example WOA algorithm on F4 functions.

Average EF criterion

In this section, average values EF are shown in table 9. The first column indicates the average values for the unimodal function, in which the SSA method has the highest value with 71%. This means that the SSA method has provided the highest value in obtaining the global optimum answer. In this column, the DA method has the weakest performance to obtain the global answer with only 0.28%. The second column indicates the performance of the methods in multi-modal functions. In this column, the WOA method has the best performance with 49.83%. Also, the DA method is known as the weakest method with 0.50%. The third column is calculated for fixed-dimension multimodal functions. In this column, the SSA method results in the best performance with 30.6%. Also, the DA method provides the lowest value with 8.80%.

Table 9: Average EN criterion

	Function type		
Algorithm	**Unimodal**	**Multi modal**	**Fixed-dimension multimodal**
WOA	28.57	**49.83**	22.60
GWO	58.00	46.66	28.60
DA	0.28	0.50	8.80
SSA	**71**	35.33	**30.6**

Average run time

One of the important matters in engineering discussions is to obtain the desired answer in the shortest time. Thus, the average run time of each algorithm is investigated in Table 10. The average time is calculated on one type of benchmark function to test the powerfulness of each method in run time. In the first column, the average run time on the unimodal functions is calculated that includes F1-F7 functions. The fastest algorithm is the WOA algorithm with 0.1091 seconds while the DA method is the slowest method with 9.6333 seconds. In the multi-modal functions that include F8-F13 functions, the WOA algorithm is the best algorithm with 0.1267 seconds, and the DA algorithm has the weakest performance with 12.6949 seconds. In the third column, the fixed-dimension multimodal functions that include F14-F23 functions are present. In this column, the WOA algorithm is the fastest algorithm with 0.0556 seconds while the DA method is the slowest method in terms of runtime with only 6.3046 seconds. The obtained results show that the DA method is the slowest algorithm in comparison with the other methods.

Table 10: Average run time (sec)

	Function type		
Algorithm	**Unimodal**	**Multi modal**	**Fixed-dimension multimodal**
WOA	**0.1091**	**0.1267**	**0.0556**
GWO	0.2483	0.2543	0.0848
DA	9.6333	12.6949	6.3046
SSA	0.1444	0.1598	0.1112

5- Conclusion

In the research performed in 2020, more than 300 meta-heuristic methods have been reported since the appearance of the genetic algorithm [29]. In the past, the number of meta-heuristic methods was limited. For example, users chose between genetic algorithms and particle swarm optimization. Nowadays users being confused about choosing the type of the method. Undoubtedly, due to the great numbers of meta-heuristic methods, it is time to write or even invent new metrics and standards to classify and rank these methods. Otherwise, we are facing with a large number of meta-heuristic algorithms that are constantly being added to their collection. This causes that the user is unable to choose an appropriate method for his problem.

In this paper, we have tried to investigate the statistical performance of WOA, GWO, DA and SSA algorithms besides defining powerfulness and effectiveness metrics. These metrics help users to compare the performance of different meta-heuristic methods. In this regard, a suitable platform and solution are provided to choose between them by the users.

In future work, we will gradually introduce new evaluation criteria in order to accurately use meta-heuristic algorithms.

References

[1] R. Bellman, "Dynamic Programming", Science, Vol. 153, No. 3731, 1966, pp. 34-37.

[2] W. Kuo, V. R. Prasad, F. A. Tillman, and C.-L. Hwang, Optimal Reliability Design: Fundamentals and Applications, Cambridge University Press, 2001.

[3] J. A. Snyman, Practical Mathematical Optimization. Springer, 2005.

[4] I. BoussaïD, J. Lepagnot, and P. Siarry, "A Survey on Optimization Metaheuristics", Information Sciences, Vol. 237, 2013, pp. 82-117.

[5] A. Sezavar, H. Farsi, and S. Mohamadzadeh, "A Modified Grasshopper Optimization Algorithm Combined with CNN for Content Based Image Retrieval", International Journal of Engineering, Vol. 32, No. 7, 2019, pp. 924-930.

[6] A. H. Hosseinian and V. Baradaran, "A Multi-Objective Multi-Agent Optimization Algorithm for the Community Detection Problem", J. Inform. Syst. Telecommun, Vol. 6, No. 1,2019, pp. 169-179.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization By Simulated Annealing", Science, Vol. 220, No. 4598, 1983, pp. 671-680.

[8] J. R. Koza and J. R. Koza, Genetic Programming, On the Programming of Computers :Natural Selection MIT press, 1992.

[9] A. Walker, J. Hallam, and D. Willshaw, "Bee-Havior in a Mobile Robot: The Construction of a Self-Organized Cognitive Map and Its Use in Robot Navigation within a Complex, Natural Environment", IEEE International Conference on Neural Networks, 1993, pp. 1451-1456.

[10] F. Glover, "Tabu Search for Nonlinear and Parametric Optimization (With Links to Genetic Algorithms)", Discrete Applied Mathematics, Vol. 49, No. 1-3, 1994, pp. 231-255.

[11] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", in Proceedings of ICNN'95-International Conference on Neural Networks, 1995, Vol. 4, pp. 1942-1948.

[12] K. M. Passino, "Biomimicry of Bacterial Foraging for Distributed Optimization and Control", IEEE control systems magazine, Vol. 22, No. 3, 2002, pp. 52-67.

[13] D. Simon, "Biogeography-Based Optimization", IEEE Transactions on Evolutionary Computation, Vol. 12, No. 6, 2008, pp. 702-713.

[14] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", Advances in Engineering Software, Vol. 69, 2014, pp. 46-61.

[15] S. Mirjalili, "The Ant Lion Optimizer", Advances in Engineering Software, Vol. 83, 2015, pp. 80-98.

[16] S. Mirjalili, "Moth-flame Optimization Algorithm: A Novel Nature-Inspired Heuristic Paradigm", Knowledge-Based Systems, Vol. 89, 2015, pp. 228-249.

[17] S. Mirjalili "Dragonfly Algorithm: a New Meta-Heuristic Optimization Technique for Solving Single-Objective, Discrete, and Multi-Objective Problems", Neural Computing and Applications, Vol. 27, No. 4, 2016, pp. 1053-1073.

[18] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-Verse Optimizer: a Nature-Inspired Algorithm for Global Optimization", Neural Computing and Applications, Vol. 27, No. 2, pp. 495-513, 2016.

[19] S. Mirjalili, "SCA: a Sine Cosine Algorithm for Solving Optimization Problems", Knowledge-Based Systems, Vol. 96, 2016, pp. 120-133.

[20] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm", Advances in Engineering Software, Vol. 95, 2016, pp. 51-67.

[21] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A Bio-Unspired Optimizer for Engineering Design Problems", Advances in Engineering Software, Vol. 114, 2017, pp. 163-191.

[22] S. M. Almufti, "Historical Survey on Metaheuristics Algorithms", International Journal of Scientific World, Vol. 7, No. 1, 2019, pp. 1.

[23] S. Shirke and R. Udayakumar, "Evaluation of Crow Search Algorithm (CSA) for Optimization in Discrete Applications", International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 584-589.

[24] M. Dorigo and G. Di Caro, "Ant Colony Optimization: a New Meta-Heuristic", in Proceedings of the Congress on Evolutionary Computation-CEC99, Vol. 2, 1999, pp. 1470-1477.

[25] M. Clerc, Particle Swarm Optimization. John Wiley & Sons, 2010.

[26] J. G. Digalakis and K. G. Margaritis, "On Benchmarking Functions for Genetic Algorithms", International Journal of Computer Mathematics, Vol. 77, No. 4, 2001, pp. 481-506.

[27] M. Molga and C. Smutnicki, "Test Functions for Optimization Needs", Test Functions for Optimization Needs, Vol. 101, 2005, pp. 48.

[28] X.-S. Yang, "Firefly Algorithm, Stochastic Test Functions and Design Optimisation," International Journal of Bio-Inspired Computation, Vol. 2, No. 2, 2010, pp. 78-84.

[29] D. Molina, J. Poyatos, J. Del Ser, S. García, A. Hussain, and F. Herrera, "Comprehensive Taxonomies of Nature-and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations", Cognitive Computation, Vol. 12, No. 5, 2020, pp. 897-9339.

Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 61-67

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Deep Learning Approach for Cardiac MRI Images

Afshin Sandooghdar[1], Farzin Yaghmaee [1*]

[1]. Electrical and Computer Engineering Department, Semnan University, Semnan, Iran

Received: 22 May 2021 / Revised: 08 Aug 2021/ Accepted: 02 Sep 2021
DOI: https://doi.org/10.52547/jist.16121.10.37.61

Abstract

Deep Learning (DL) is the most widely used image-analysis process, especially in medical image processing. Though DL has entered image processing to solve Machine Learning (ML) problems, identifying the most suitable model based on evaluation of the epochs is still an open question for scholars in the field. There are so many types of function approximators like Decision Tree, Gaussian Processes and Deep Learning, used in multi-layered Neural Networks (NNs), which should be evaluated to determine their effectiveness. Therefore, this study aimed to assess an approach based on DL techniques for modern medical imaging methods according to Magnetic Resonance Imaging (MRI) segmentation. To do so, an experiment with a random sampling approach was conducted. One hundred patient cases were used in this study for training, validation, and testing. The method used in this study was based on full automatic processing of segmentation and disease classification based on MRI images. U-Net structure was used for the segmentation process, with the use of cardiac Right Ventricular Cavity (RVC), Left Ventricular Cavity (LVC), Left Ventricular Myocardium (LVM), and information extracted from the segmentation step. With train and using random forest classifier, and Multilayer Perceptron (MLP), the task of predicting the pathologic target class was conducted. Segmentation extracted information was in the form of comprehensive features handcrafted to reflect demonstrative clinical strategies. Our study suggests 92% test accuracy for cardiac MRI image segmentation and classification. As for the MLP ensemble, and for the random forest, test accuracy was equal to 91% and 90%, respectively. This study has implications for scholars in the field of medical image processing.

Keywords: Deep Learning; Neural Networks; Magnetic Resonance Imaging (MRI); Disease Prediction.

1- Introduction

According to a World Health Organization (WHO) report in 2016, approximately 17.9 people have deceased; due to cardiovascular disease. [1] To avoid the growing number of deaths caused by cardiovascular disease, diagnosis of the disease has attracted the attention of physicians. At present, physicians diagnose cardiovascular disease based on the patient's medical history and physical examinations. Their experience is also involved in interpreting and treating the disease.

The absence of computer-based processors results in biased results. In addition, there are higher chances of making a mistake in interpretation when the outcomes are non-producible. Needless to mention that such diagnosis is more expensive.

Thus, there is a need to perform an automated process on medical images that includes diagnosis and treatment with high accuracy. In some cases, there are no physicians available for doing the process of diagnosis and treatment. Factors such as labor time, subjective inclination, and intended reproducibility have hindered comprehensive quantitative estimations. This situation perpetuates the significance of Machine Learning (ML) methods with a combination of Deep Learning as a practical solution to implement automating and segmentation on medical images. Artificial Intelligence (Al) is the basis of Machine learning (ML), through which computers can learn a task using data rather than being specifically programmed.

One of the main categories in ML is deep Learning (DL). DL has neural networks designed to employ data to capture hierarchical levels of abstraction. They are made up of several stacked layers.

Convolutional Neural Networks (CNNs) have provided excellent results for segmenting medical images. [2] Convolutional Neural Network (CNN) is a subcategory of neural networks that includes a stack of layers like convolution layers, and pooling layers. Each layer does a specific action. [3] The input layer is the beginning layer that is connected to an input image directly. The number of pixels in the image is equal to the number of neurons in this layer. The next layer is the convolutional layer, which is helpful for sequential data and images. Parameters in the

✉ **Corresponding Author**
f_yaghmaee@Semnan.ac.ir

layers include a set of kernels (filters) that designers define, and they have arbitrary sizes. [4]

First layers in image classification, learn how to detect patterns, edges, and textures. Subsequent layers can detect the objects, either entirely or partially. [5] There are several types of deep neural networks for use in medical imaging. For example, the gold standard in cardiac function research is cardiac magnetic resonance imaging (CMR), which assesses the left and right ventricular ejection fractions (EF) and stroke volumes (SV), as well as the mass of the left ventricle and the thickness of the myocardium. [6]

Cardiac MRI is an imaging technique based on a non-invasive structure that can visualize the heart anatomy.it works based on radio-frequency waves and generates images of the heart. [7]

Appropriate results are not gained in clinical practices, which are performed semi-automatically with inaccurate segmentation. In addition, it will be very time-consuming. The accuracy and speed in the semi-automatic method are lower compared to the automatic method. [8][9]

MRI images have several problems that are fully expressed in [10]. Here, we briefly mention a few of them

1. Intensity and form variability of cardiac functions in patients with various pathologies.
2. Trabeculae and papillary muscles with intensities equivalent to the myocardium are present.
3. There is a lack of contrast between the myocardium and its surroundings.
4. Blood flow induces brightness heterogeneities in the left and right ventricular cavities.

In the years before 2012, various works have been done on segmenting MRI images that have used methods other than deep learning. In these articles, various datasets have been used, which only included the ground truth for the ED and ES ventricular volume. [11][12][13]

This paper has used a data set called automatic Cardiac Diagnosis Challenge (ACDC), which is a public data set. Compared to previous datasets, ACDC has a more extensive scope on cardiac. It has manual expert segmentation of the Left and Right Ventricle (LV/RV).

The main objective of this paper is to evaluate convolutional neural networks (CNNs) based on U-Net structure on the ACDC data set for segmentation and classification of cardiac images.

Using the method presented in this paper, Computer-Aided Diagnosis (CAD) is possible for patients with myocardial infarction.

In the next section, we evaluate MRI segmentation methods and mention related works based on deep learning techniques. In the following, we will explain our method. In the end, the results of research and implementation based on U-Net are presented. In this section, the idea that we are currently working on is also presented.

In the rest of this paper, we discuss related work in section 2, evaluation method in section 3, and results in section 4. Finally, we discuss conclusions and the future works with using deep learning methods in medical image processing.

2- Related Works

Certainly, before developing methods based on deep learning, the usual methods in image processing were used to segment images, in the following, some of which are mentioned.

In 2011, Petitjean et al. presented a comprehensive analysis of segmentation methods for defining the LV/RV in short-axis cardiac MR images. [14]

In this paper, the authors have published the results of about 70 articles in this field. The presented methods can be divided into two parts, which are weak and strong prior methods. Weak prior includes weak assumptions like anatomical information or spatial intensity.

The first category includes techniques based on image, like, threshold, and dynamic programming. [15], classification methods based on pixels, like clustering and Gaussian mixture model fitting. [16], active contour and level set [17] and finally approaches based on the graph. [18]

The second category includes methods based on strong prior, which includes atlas-based methods [19] and active shape and appearance methods. [20] All the mentioned cases need training data set with manual annotations. In the following, we review the cases that have used deep learning techniques.

Most of the works in this section are based on segmenting cardiac MRI (e.g., Left and Right Ventricle and Left Atrium). This is indeed a limitation caused by limited data sets.

The first one that used a Fully Convolutional Network (FCN) [21] for segmentation of LV and RV on MRI images was Tran [22]. Tran and his colleagues received very good results based on FCN and performed much better than the old methods in the case of speed and accuracy. In recent years, many researchers have worked on the FCN network. Most of their focus has been on improving the network to increase segmentation capacity and greater accuracy. As example, we can mention [23], [24] and [25]. Tran studied weighted cross-entropy, weighted Dice loss, deep supervision loss, and focal loss to enhance segmentation accuracy. As a result of the low resolution and motion of cardiac MRI scanners, these loss functions work based on 2D networks rather than 3D networks for segmentation. [26][27]

Employing 2D networks for cardiac segmentation has shortcomings. Not only do they operate slice by slice, but also, they ignore inter-slice dependencies. As a result, locating and segmenting the heart on difficult slices like apical and basal slices, where the ventricle contours aren't well defined, is challenging. Various works have attempted to resolve this problem by adding additional contextual details to direct 2D FCN. [28]

Other researchers work based on extract spatial information from adjacent slices for doing better segmentation using Recurrent Neural Networks (RNNs) [22], [29]. Based on our research, very few papers have been done on atrial segmentation. Table 1 shows the summary of the mentioned papers prepared by Chen and his colleagues. [2]

Table 1: Deep learning methods on cardiac MRI segmentation. S.A.: Short-Axis view prepared by Chen and colleagues

Selected works FCN- based	Description	Types of Images	Structure
Tarn(2016)	2D FCN	**S.A.**	Bi-Ventricle
Jang et al. (2017)	2D M-Net with weighted cross entropy	**S.A.**	Bi-Ventricle
Khened (2019)	2D U-net	**S.A.**	Bi-Ventricle
Fahmy (2019)	2D FCN	**S.A.**	LV
Poudel (2016)	2D FCN + RNN	**S.A.**	Bi-Ventricle
Patravali (2017)	2D Multi Channel FCN	**S.A.**	Bi-Ventricle

3- Methodology

U-Net is a well-known structure in segmenting medical images. This network was provided by Ronneberger et al. [30] based on the deconvolution concept presented by [31]. FCN inspires the U-Net model. Furthermore, U-Net has a depth of 19 layers. The superior r design of skip connections between different stages of the network is one of the benefits of the U-Net. [32] It is vital to overcome the trade-off, as it is a negative concept in U-Net. Trade-off reduces localization caused by the lack of pooling layers. The pooling layers are required for large-sized patches.

In order to overcome the trade-off between localization and the use of context, U-Net is modified. [2] The main modification is the shortcut connections between the layers with equal resolution. The high-resolution of the deconvolution layers is the result of these connections.

U-Net is designed for semantic segmentation. The role of semantic segmentation is to group sections of an image that belong to the same object class together. U-Net has contracted and expansive paths. The main feature of the contracting path is the convolutional network. This network has repeated application of two 3x3 convolutions, which are followed by a Rectified Linear Unit (ReLU) and a 2x2 max pooling operation with stride 2 for down-sampling.

In order to decrease the storage and difficulties of transmitting images, the spatial resolution should be reduced. At the same time, the same 2D representation is kept. This process is known as down sampling. In addition, the number of feature channels should be doubled at each down sampling step.

The expansive path, on the other hand, entails an up-sampling of the feature map. It has a 2x2 convolution ("up-convolution") designed to cut the number of feature channels into halves.

It is important to up-sample the resolution of the feature map. The main approaches used for this purpose are down-sampling the resolution by summarizing a local area with a single value (average or max pooling), "un-pooling" operations, up-sample the resolution by distributing a single value into a higher resolution, and pooling operations.

In order to avoid the loss of border pixels, cropping should be implemented in convolutions. Thus, a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes at the final convolution. The U-Net has a total of 23 convolutional layers which for a network with a decoder and encoder structure. This structure is depicted in Figure 1.

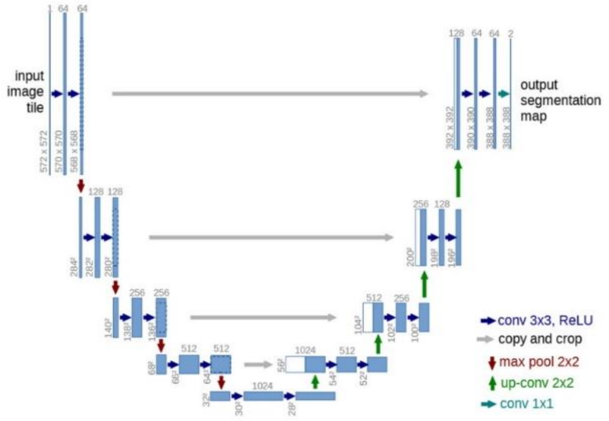


Fig. 1 The structure of the U-Net [29].

In this paper, we evaluated the U-Net structure with the ACDC Data set. Our goal is to measure the accuracy of segmentation and classification of MRI images with this network because one advantage of using this network is that it can provide acceptable results with a small amount of data. Our data set include MRI images for 150 patients.

The ACDC using two types of scanners with different magnetic strengths, one 1.5T and the second 3.0T. Each time series has between 28 and 40 3D volumes that cover the cardiac cycle. These volumes are segmented manually by clinical experts on end-diastolic (ED) and end-systolic (ES) phase instants done on LVC, LVM, and RVC. ACDC dataset includes five groups. Except for one of the healthy patients, four groups refer to pathological patients. These healthy patients are those who have had a myocardial infarction (MINF), dilated cardiomyopathy (DCM), hypertrophic cardiomyopathy (HCM), abnormal right ventricle (ARV), and healthy or normal subjects (NOR). The height and weight of patients give more information about this status. The dataset is part into 100 training and 50 test patients. Division and classification ground truth is given as it were for the 100 training cases.

In this paper, domain-specific features extracted in each time step of MRI based on multi-structure segmentation, which are stimulated by the workflow of a cardiologist, to then train an ensemble of classifiers for disease prediction. Figure 2.

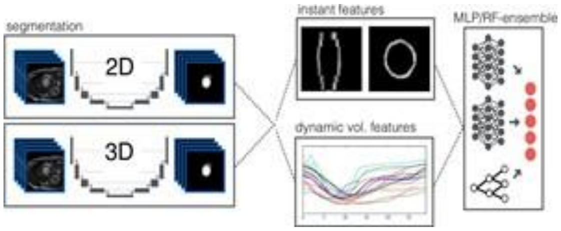


Fig. 2 Outline of the using method. Model-based on 2D and 3D from segmentation process averaged and then after extracting dynamic volume and instant features, used for disease prediction based on the ensemble of a classifier.

The methods used in this article are segmenting the images and classification based on the properties extracted from segmentation. These processes are explained below. Figure 3.

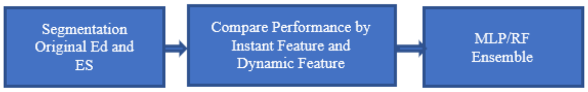

Segmentation Original Ed and ES
Compare Performance by Instant Feature and Dynamic Feature
MLP/RF Ensemble

Fig. 3 Base Flowchart

3-1- Segmentation

We used the U-net network to segment the images [29]. In this way, we have made changes in the network, and it can be used for 2D and 3D images.

The 3D segmentation demonstrates a context aggregation pathway through a localization pathway. These two ways at various scales are interconnected. This is because we can combine context features with the corresponding local information Figure 4.

Operations like upscaling and pooling are performed in the x-y plane because of the low resolution of input (z-resolution). The setting within the z-dimension is exclusively totaled through the 3D convolutions.

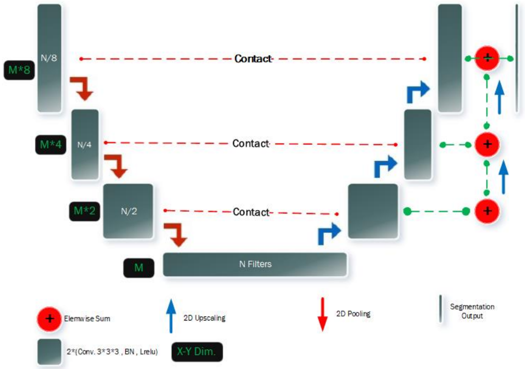


Fig. 4 Architecture of segmentation network based on 3 Dimensional, for the 2D size of patch is 352 × 352 and uses 48 initial features and 2Dimentional convolutions. Gray blocks are features that extracted and each of them includes ReLU, by batch normalization

It is mentioned that, Because of network layers that include 18 layers, there are no residual connections.

Upscaling (pooling) operations are responsible for halved 26 feature maps. In fact, there are four upscaling operations, and each of them doubled the initial number of twenty-six feature maps, so, resulting in a maximum of 416 feature maps at the bottom of the U-net structure.

Our training 3D U-Net is based on 300 epochs in 5-fold cross-validation that include a pixel-wise categorical cross-entropy loss. The 2D network is the same as the 3D network with 2D convolutions. Obviously, a 2D network requires less memory, so we use 48 initial feature maps.

For the training network, we use, batch size of 10 with input patches with a size of 352*352 and a multiclass dice loss, Eq. (1)

$$\mathrm{L_{dc}} = -\frac{2}{|\mathrm{k}|}\sum_{\mathrm{k}\in\mathrm{K}}\frac{\sum_{\mathrm{i}}\mathrm{u_i^k v_i^k}}{\sum_{\mathrm{i}}\mathrm{u_i^k}+\sum_{\mathrm{i}}\mathrm{v_i^k}} \qquad (1)$$

shows the SoftMax output of the network, encoding of Ground Truth (GT) segmentation maps denotes with v. Both u and v are of size i × k with i being the number of pixels in the training patch and $\mathrm{k} \in \mathrm{K}$ being the classes.

3-2- Classification

After the segmentation process, we can obtain two sets of features. These features are used for the classification of disease. Table 2 shows features of these two sets.

Table 2: Both Ed and ES used for extracted instant features.

Instant Features	LVM	LVC	RVC
Maximum Thickness	●	-	-
Minimum Thickness	●	-	-
Std. Thickness	●	-	-
Mean Thickness	●	-	-
Mass	●	-	-
Maximum Circumference	●	-	●
Mean Circumference	●	-	●
Mean Circularity	●	-	●

Dynamic Volume Features

Dynamic Volume Features	LVM , LVC , RVC
Maximum Volume	Covers All Features (●)
Minimum Volume	
Dynamic Ejection Fraction	
Volume Median	
Volume Standard Deviation	
Volume Skewness	
Volume Kurtosis	

Instant Features include information of local and global shape, which global information include circumference, circularity, the thickness of LVM, and so on, and local information including, size of RVC at the apex, LVM thickness between RVC and LVC, etc.

For Dynamic Volume Features, the segmentation process is done to specify anatomical structures in all time steps in the MRI.

For the classification step, an ensemble of 50 MLP trained based on features of Table 2.

Pathology classification with a random forest was also used.

The number of hidden layers in the Structure of Multilayer perceptron is 4. 32 units are also used for training in each layer and batch normalization, ReLU with a Gaussian noise layer (σ = 0.1) used in training. We divided our data set into two parts, 75 percent for training selected randomly and 25 percent of the dataset for epoch selection. Multilayer perceptron trained based on 400 epochs. We trained a random forest is trained with Thousand trees. Throughout Multilayer Perceptron score, can be obtained During testing, with the SoftMax outputs of all MLPs were averaged, which was recombined in this way with the arbitrary random forest yield to get the final ensemble prediction.

3-3- Experimental Analysis

The impact of the proposed network architecture, loss function, data-augmentation scheme, and the influence of Region of Interest (ROI) cropping and post-processing was measured in this study through an experiment. Similar to many studies that use the ACDC training dataset, the unit of the evaluation was Dice score and Hausdor Distance (HD) in mm. Also, the TensorFlow software was employed to design the neural networks. A desktop computer with the following specifications was used to run the experiment.

GPU: NVIDIA-Titan-X GPU, CPU: Intel Core i7-4930K 12-core CPUs @ 3.40GHz, RAM: 64GB

3-4- ACDC Dataset

The dataset used in this study entailed 100 cases of patients (1:8k 2D images). This data set was divided into 70 cases for training, 15 cases for validation, and 15 cases for testing. A random sampling method, i.e., stratified sampling, was employed. Stratified sampling can be employed when the total number of cases is known and can ensure that each stratum entails an equal number of cases from different cardiac disease groups. Each case had approximately 20 2D images with ground-truth annotations for LV, RV, and MYO at the ED and ES phases.

3-5- Training

The researcher used a training patch with 16 ROI cropped 2D images. The images had 128x128 dimensions. This network was trained for 200 epochs.

In order to select a suitable model, the models were evaluated after each epoch. Focus was accorded to the highest Dice score for MYO class on the validation set. Figure 5 is an example of the intermediate feature maps of the trained network in the ACDC-2017 dataset.

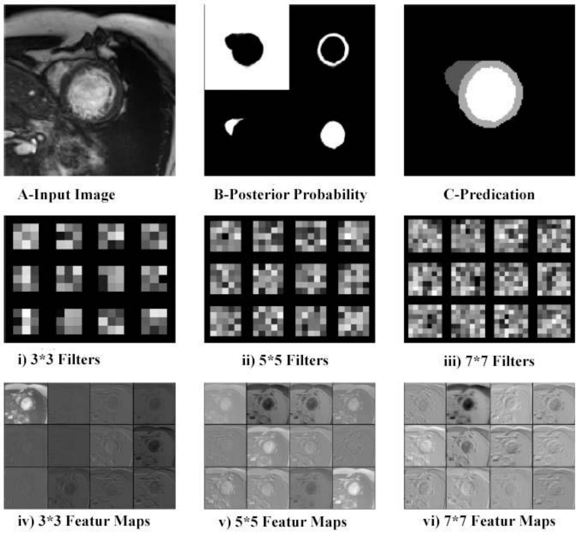


Fig. 5 The figures compare illustrate the feature maps of the trained model. (A) Input image fed to the network, (B) posterior probability maps after soft-max output, (C) The final prediction of labels, (i) - (iii) visualization of the initial layers kernels- 3x3, 5x5 and 7x7, (iv) - (vi) Filter response to the input image (A).

4- Results

The results of the five-fold Cross Validation (CV) are shown in Table 3. The results include Hausdorff distances. Individual dice scores are also mentioned in Table 3.

Table 3: Segmentation process with Dice scores and Hausdorff distances for five-fold CV

	Case	Dice			Hausdorff (mm)		
		RCV	LMV	LVC	RVC	LVM	LVC
DCM	ED	0.940	0.903	0.967	20.86	8.20	7.115
	ES	0.869	0.912	0.914	17.941	8.160	5.886
	Result	**0.906**	**0.909**	**0.943**	**19.828**	**8.154**	**6.543**
HCM	ED	0.937	0.902	0.967	12.720	8.706	7.255
	EC	0.876	0.905	0.934	18.325	11.353	14.514
	Result	**0.909**	**0.904**	**0.951**	**15.322**	**10.104**	**11.023**
MINF	ED	0.936	0.894	0.963	13.383	9.631	6.880
	EC	0.888	0.904	0.905	18.636	11.75	9.598
	Result	**0.912**	**0.902**	**0.933**	**16.106**	**10.732**	**8.115**
NOR	ED	0.938	0.886	0.972	9.764	7.231	4.624
	ES	0.884	0.901	0.944	11.407	9.164	7.665
	Result	**0.911**	**0.898**	**0.955**	**10.612**	**8.398**	**6.332**
RV	ED	0.946	0.908	0.963	14.725	10.716	9.393
	ES	0.853	0.914	0.921	15.124	11.768	11.225
	Result	**0.901**	**0.9101**	**0.944**	**15.131**	**11.608**	**10.623**

Compared to the 2D model, the 3D model provided better results. The results can be seen in Table 4.

Table 4: Results from CV on the training set are shown for the 2D and 3D model

		Dice			Hausdorff (mm)		
		RVC	LVM	LVC	RVC	LVM	LVC
CV	2D	0.905	0.904	0.946	14.292	8.898	7.053
	3D	0.878	0.874	0.926	16.288	10.435	9.776
	Result	0.909	0.904	0.946	15.292	9.666	8.414
TEST	Result	0.926	0.912	0.954	11.133	8.698	7.142

The dice score is a statistical procedure employed to measure the similarity of two samples. On the contrary, Hausdorff gauges the difference between two subsets of a metric space. For classification, we train classification ensemble with features extracted from the segmentation step, the test accuracy of classification was 92 percent. For the MLP ensemble, 91 percent was achieved and for the random forest was 90%. Table 5 indicated confusion matrices.

Table 5: Error matrices or confusion matrices of the ensemble predictions from CV showed training and test set. the predicted class is based on Rows and the target class shows in columns

NOR	18	0	1	0	1
DCM	0	19	0	1	0
HCM	0	0	19	1	0
MNF	0	1	0	19	0
RVA	1	0	0	0	19
	NOR	DCM	HCM	MNF	RVA

NOR	10	0	0	0	0
DCM	0	9	0	1	0
HCM	1	0	9	0	0
MNF	0	2	0	8	0
RVA	0	0	0	0	10
	NOR	DCM	HCM	MNF	RVA

5- Conclusion and Future Works:

In this paper, we proposed a fully automatic process for MRI image segmentation and classification .in the first part of our method, we developed segmentation architecture based on U-Net structure and trained the network on ED and ES. The first part can be done on 3D and 2D images.

Features extraction is the output of the segmentation step and is used in the second step (ensemble classifier) to predict the diagnosis. The test accuracy of classification was 92 percent. For the MLP ensemble, 91 percent was achieved and for the random forest was 90%. Table 5 indicated confusion matrices.

Also in this paper, We developed a precise multi-structural segmentation method that is taught only in the case of ED and ES phases. Our approach revolves around the use of both two-dimensional and three-dimensional models and uses the corresponding benefits through combination. The results are strong against incisional inconsistencies, different magnetic resonance imaging protocols of the heart as well as various injuries. We scored 0.954 (LVC), 0.926 (RVC), and 0.912 (LVM) on the ACDC test suite.

The findings of the current study can increase the outcome of computer image processors and enhance science in the arena of computer-aided diagnosis. Physicians need for semi-automated or automated processors can be fulfilled by increasing the production of digital information. The use of computer-based image processors results in quick production of robust and accurate data. This is among the main contributions of this study.

Future work

Machine learning is divided into three parts: supervised Learning (SL), unsupervised learning (UL), and reinforcement learning [2]. Deep Learning (DL), which uses multi-layered neural networks, is used in all three parts of machine learning. The difference between Deep Reinforcement Learning (DRL) and other machine learning methods is using the agent to perform the task in the environment. A very interesting way to segment MRI images and classification tasks is to use Deep Reinforcement Learning methods. The use of additional agents can also be used to classify and segment images.

References

[1] “Automated Cardiac Diagnosis Challenge.” https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[2] C. Chen et al., “Deep learning for cardiac image segmentation: A review,” arXiv, 2019, doi: 10.3389/fcvm.2020.00025.

[3] M. H. Hesamian, W. Jia, X. He, and P. Kennedy, “Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges,” J. Digit. Imaging, vol. 32, no. 4, pp. 582–596, 2019, doi: 10.1007/s10278-019-00227-x.

[4] M. Samieiyeganeh, R. W. B. O. K. Rahmat, F. B. Khalid, and K. A. Bin Kasmiran, “An overview of deep learning techniques in echocardiography image segmentation,” Journal of Theoretical and Applied Information Technology, vol. 98, no. 22. pp. 3561–3572, 2020.

[5] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, “An introduction to deep reinforcement learning,” Foundations and Trends in Machine Learning, vol. 11, no. 3–4. pp. 219–354, 2018, doi: 10.1561/2200000071.

[6] O. Bernard et al., “Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved?,” IEEE Trans. Med. Imaging, vol. 37, no. 11, pp. 2514–2525, 2018, doi: 10.1109/TMI.2018.2837502.

[7] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on MRI,” arXiv. 2018.

[8] C. A. Miller et al., “Quantification of left ventricular indices from SSFP cine imaging: Impact of real-world variability in analysis methodology and utility of geometric modeling,” J. Magn. Reson. Imaging, vol. 37, no. 5, pp. 1213–1222, 2013, doi: 10.1002/jmri.23892.

[9] M. Khened, V. A. Kollerathu, and G. Krishnamurthi, “Fully convolutional multi-scale residual DenseNets for cardiac segmentation and automated cardiac diagnosis using

ensemble of classifiers," Med. Image Anal., vol. 51, pp. 21–45, 2019, doi: 10.1016/j.media.2018.10.004.

[10] Q. Zheng, H. Delingette, N. Duchateau, and N. Ayache, "3-D Consistent and Robust Segmentation of Cardiac Images by Deep Learning With Spatial Propagation," IEEE Trans. Med. Imaging, vol. 37, no. 9, pp. 2137–2148, 2018, doi: 10.1109/TMI.2018.2820742.

[11] A. Suinesiaputra et al., "from Cardiac MRI: A Collation Study." pp. 88–97, 2012.

[12] C. Petitjean et al., "Right ventricle segmentation from cardiac MRI: A collation study," Medical Image Analysis, vol. 19, no. 1. pp. 187–202, 2015, doi: 10.1016/j.media.2014.10.004.

[13] X. Zhuang et al., "Cardiac Segmentation on Late Gadolinium Enhancement MRI: A Benchmark Study from Multi-Sequence Cardiac MR Segmentation Challenge," no. Mi, 2020, [Online]. Available: http://arxiv.org/abs/2006.12434.

[14] C. Petitjean and J. N. Dacher, "A review of segmentation methods in short axis cardiac MR images," Medical Image Analysis, vol. 15, no. 2. pp. 169–184, 2011, doi: 10.1016/j.media. 2010.12.004.

[15] L. M. C. Leon, K. C. Ciesielski, and P. A. V. Miranda, "Efficient Hierarchical Multi-Object Segmentation in Layered Graphs," Math. Morphol. - Theory Appl., vol. 5, no. 1, pp. 21–42, 2021, doi: 10.1515/mathm-2020-0108.

[16] H. Wei, W. Xue, and D. Ni, "Left ventricle segmentation and quantification with attention-enhanced segmentation and shape correction," ACM Int. Conf. Proceeding Ser., pp. 226–230, 2019, doi: 10.1145/3364836.3364881.

[17] I. Symposium and B. Imaging, "SEGMENTING THE LEFT VENTRICLE IN CARDIAC IN CARDIAC MRI : FROM HANDCRAFTED TO DEEP REGION BASED DESCRIPTORS Daniela O . Medley , Carlos Santiago , and Jacinto C . Nascimento Institute for Systems and Robotics , Instituto Superior T ´ ecnico , Lisbon , Por," no. Isbi, pp. 644–648, 2019.

[18] W. Wang, Y. Wang, Y. Wu, T. Lin, S. Li, and B. Chen, "Quantification of Full Left Ventricular Metrics via Deep Regression Learning with Contour-Guidance," IEEE Access, vol. 7, pp. 47918–47928, 2019, doi: 10.1109/ACCESS.2019.2907564.

[19] W. Bai, W. Shi, C. Ledig, and D. Rueckert, "Multi-atlas segmentation with augmented features for cardiac MR images," Medical Image Analysis, vol. 19, no. 1. pp. 98–109, 2015, doi: 10.1016/j.media.2014.09.005.

[20] S. Moradi et al., "MFP-Unet: A novel deep learning based approach for left ventricle segmentation in echocardiography," Phys. Medica, vol. 67, no. 110, pp. 58–69, 2019, doi: 10.1016/j.ejmp.2019.10.001.

[21] E. Shelhamer, J. Long, and T. Darrell, "IEEE Transactions on Pattern Analysis and Machine Intelligence Fully Convolutional Networks for Semantic Segmentation." [Online]. Available: http://www.ieee.org/publications_standards/publications/rights/index.html.

[22] R. P. K. Poudel, P. Lamata, and G. Montana, "Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10129 LNCS. pp. 83–94, 2017, doi: 10.1007/978-3-319-52280-7_8.

[23] Y. Jang, Y. Hong, S. Ha, S. Kim, and H. J. Chang, "Automatic segmentation of LV and RV in cardiac MRI," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10663 LNCS, pp. 161–169, 2018, doi: 10.1007/978-3-319-75541-0_17.

[24] J. Duan et al., "Automatic 3D Bi-Ventricular Segmentation of Cardiac Images by a Shape-Refined Multi- Task Deep Learning Approach," IEEE Trans. Med. Imaging, vol. 38, no. 9, pp. 2151–2164, 2019, doi: 10.1109/TMI.2019.2894322.

[25] A. S. Fahmy, H. El-Rewaidy, M. Nezafat, S. Nakamori, and R. Nezafat, "Automated Analysis of Myocardial Native T1 Mapping Images Using Fully Convolutional Neural Networks," Journal of Cardiovascular Magnetic Resonance, vol. 21, no. 7. pp. 1–12, 2019.

[26] M. Kendirci, S. Nowfar, and W. J. G. Hellstrom, "The impact of vascular risk factors on erectile function," Timely topics in medicine. Cardiovascular diseases, vol. 9. p. E11, 2005.

[27] E. Ermis et al., "The relationship between erectile dysfunction and the Atherogenic Index of Plasma," Int. J. Impot. Res., vol. 32, no. 4, pp. 462–468, 2020, doi: 10.1038/s41443-019-0167-2.

[28] C. Zotti, Z. Luo, A. Lalande, and P. M. Jodoin, "Convolutional Neural Network with Shape Prior Applied to Cardiac MRI Segmentation," IEEE J. Biomed. Heal. Informatics, vol. 23, no. 3, pp. 1119–1128, May 2019, doi: 10.1109/JBHI.2018.2865450.

[29] J. Patravali, S. Jain, and S. Chilamkurthy, "2D-3D fully convolutional neural networks for cardiac MR segmentation," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10663 LNCS. pp. 130–139, 2018, doi: 10.1007/978-3-319-75541-0_14.

[30] S. Jahangard and M. Shahedi, "U-Net Based Architecture for an Improved Multiresolution Segmentation in Medical Images," pp. 1–22, 2020.

[31] F. B. M. Suah, "Preparation and characterization of a novel Co(II) optode based on polymer inclusion membrane," Analytical Chemistry Research, vol. 12. pp. 40–46, 2017, doi: 10.1016/j.ancr.2017.02.001.

[32] P. F. Christ et al., "Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9901 LNCS. pp. 415–423, 2016, doi: 10.1007/978-3-319-46723-8_48.

Journal of Information Systems and Telecommunication
Vol.10, No.1, January-March 2022, 68-79

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

A Survey on Multi-document Summarization and Domain-Oriented Approaches

Mahsa Afsharizadeh[1], Hossein Ebrahimpour-Komleh[1*], Ayoub Bagheri[2], Grzegorz Chrupala[3]

[1]. Department of Computer and Electrical Engineering, University of Kashan, Iran
[2]. Department of Social Sciences, Utrecht University, Utrecht, The Netherlands
[3]. Department of Cognitive Science and Artificial Intelligence, Tilburg University, The Netherlands

Received: 01 Jun 2021 / Revised: 06 Sep 2021/ Accepted: 06 Oct 2021
DOI: https://doi.org/10.52547/jist.16245.10.37.68

Abstract

Before the advent of the World Wide Web, lack of information was a problem. But with the advent of the web today, we are faced with an explosive amount of information in every area of search. This extra information is troublesome and prevents a quick and correct decision. This is the problem of information overload. Multi-document summarization is an important solution for this problem by producing a brief summary containing the most important information from a set of documents in a short time. This summary should preserve the main concepts of the documents. When the input documents are related to a specific domain, for example, medicine or law, summarization faces more challenges. Domain-oriented summarization methods use special characteristics related to that domain to generate summaries. This paper introduces the purpose of multi-document summarization systems and discusses domain-oriented approaches. Various methods have been proposed by researchers for multi-document summarization. This survey reviews the categorizations that authors have made on multi-document summarization methods. We also categorize the multi-document summarization methods into six categories: machine learning, clustering, graph, Latent Dirichlet Allocation (LDA), optimization, and deep learning. We review the different methods presented in each of these groups. We also compare the advantages and disadvantages of these groups. We have discussed the standard datasets used in this field, evaluation measures, challenges and recommendations.

Keywords: Multi-document Summarization; Single Document Summarization; Extractive; Abstractive; Domain-Oriented; ROUGE.

1- Introduction

Unlike in the past, there was often not enough data on every issue, but today we are faced with the issue of information overload. Obtaining the most important information from a huge amount of data is a time-consuming and difficult task. Various fields such as natural language processing (NLP), text mining and artificial intelligence were used to provide a solution to this problem. Automatic text document summarization is an important solution to this problem. Text summarization obtains a short and compact gist that preserves the main concepts of the original text. So the user can understand the concept of a long text in the form of a brief summary.

According to different perspectives, text summarization methods can be divided into several categories.

From the summary producing perspective, summarization is divided into two categories: extractive and abstractive. Extractive summarizing merges the extracts from the original text and presents them as summaries. While abstractive summarizing paraphrases the text and generates new sentences [1]. Summarization can be generic or query-based. The generic summary is about the whole text while the query-based summary is about the query being asked [2]. In terms of the number of texts, the summarization is divided into two categories: single document summarization (SDS) and multi-document summarization (MDS). The purpose of summarizing a single document is to produce a summary of one text, while the purpose of multi-document summarizing is to produce a short, relevant summary of a set of several textual documents related to a similar topic [3]. In terms of the domain of the input text, summarization techniques are divided into two categories: domain-oriented and domain-independent techniques. Domain-oriented methods perform summarization for texts related to a specific domain, for example, medicine, law, etc. While domain-

✉ **Corresponding Author**
ebrahimpour@kashanu.ac.ir

independent summarization techniques produce summaries without considering the domain of the input text.

For many years, major research was performed on SDS. Research in this field is still ongoing. MDS is necessary to apply summarization at larger scales. The following example illustrates the need for MDS. A user searches for a specific topic on the World Wide Web. This search will lead to the retrieval a lot of related documents. There is probably a lot of similar information between these related documents. In this situation, the performance of a single document summarizer on each of these documents leads to the production of multiple summaries with plenty of redundant information [4]. Therefore, a single document summarizer cannot fulfill the main goal of the summarization task, which is generating a summary with minimum redundancy and maximum relevancy [5]. Multi-document summarization has emerged as an effective solution to such a situation.

Research in the field of automatic text summarization began with SDS and moved to MDS after a while. In recent years, various approaches to MDS have been proposed.

Sometimes text documents are related to a specific domain, such as medicine, law, terrorism, etc. Applying the usual text summarization methods to such documents will not produce satisfactory results. Because text documents that are written in a particular domain often have a certain structure and characteristics. Domain-oriented summarizers encounter more challenges. Such systems, in addition to commonly used summarization techniques, use the structure and specific characteristics of that domain to identify deeper information from text documents, which results in generating more efficient summaries.

This paper is organized as follows: section 2 presents definitions, applications, and categorizations of multi-document summarizers by different authors. Section 3 provides a new categorization for MDS methods in six categories: machine learning, clustering, graph, LDA, optimization, and deep learning-based approaches. Section 4 describes domain-oriented summarization. Section 5 presents the datasets and standard measures for evaluating a summarization system. Section 6 explains challenges and recommendations in the field of MDS. Finally, section 7 concludes the paper.

2- Multi-document Summarizer

With the increase in the amount of information on web pages, finding the desired information has become a difficult issue. For a given topic, there may be hundreds of documents that are not necessarily related to it. In order to find the related information, a user needs to search among all the documents. This results in a huge amount of information, with a lot of time and effort. To cope with this problem, automatic text summarization plays a vital role [6]. Automatic summarization of text documents is a method for producing a compressed version of the original document.

So far different categorizations have been presented in the context of text summarization. Yosefi-Azar and Hamey categorized text summarization techniques into three types [7]:

- Classic approaches
- Machine learning-based approaches
- Artificial neural network-based approaches.

The initial summarization methods in the classical approaches were based on the frequency of words occurring in the text. The sum of the frequencies of the words that make up a sentence can be considered as a score to indicate the importance of that sentence in the whole text. In this set of methods, other word level and sentence level features like key phrases (for example title and heading words) and the position of sentences were also used. Cluster-based and graph-based methods can also be considered as classical methods.

Machine learning-based summarizers are trainable systems which learn how to tune their parameters to extract salient content. The set of features in these techniques is often the same as the classical methods. Machine learning summarizers include several methods such as Decision Tree, Hidden Markov Model (HMM), Support Vector Machine (SVM), and Support Vector Regression (SVR).

Artificial Neural Networks (ANN) are the other trainable systems that simulate the structure of the human brain. A neural network can be trained to learn the features of the informative sentences. Different types of shallow and deep neural networks are used to summarize the text. Deep neural networks have promising results in summarizing the text.

Maria Fuentes Fort describes the methods used in automatic summarizing from two perspectives, classical perspective and multi-task perspective [8]. Summarization categorization according to Fuentes Fort’s view is shown in Fig 1.

In the classical perspective, automatic summarization can be divided into three levels based on the level that the summarizer processes texts: surface level, entity level, and discourse level.

- **Surface Level:** Surface level techniques use shallow linguistic features to display information. The combination of these features produces a salience function for distinguishing important text information. Shallow linguistic features used in surface-based methods can be divided into four groups: term frequency-based features, location-based features, bias-based features, and cue words features.
- **Entity-Level:** Entity-level summarization methods provide an internal representation from the text to determine the salience by modeling the text entities (for example simple words, compound words, named entities, and terms) and relationships (such as similarity, proximity, co-occurrence, co-reference, etc.).
- **Discourse Level:** Discourse level methods model the global structure of the text. Cohesion and coherence are two main features in text discourse structure.

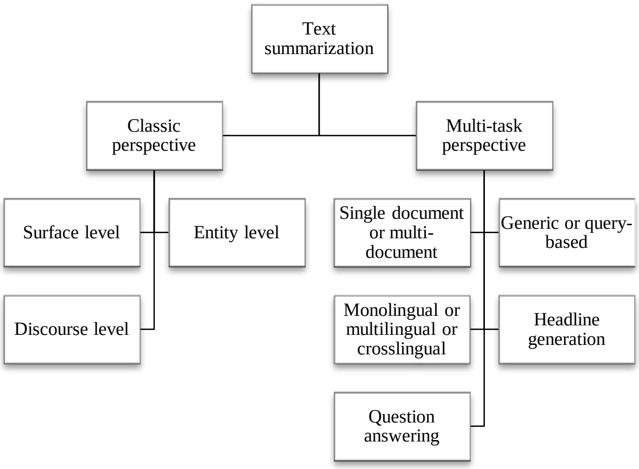


Fig. 1 Summarization categorization according to Fuentes Fort's view [8]

- Multi-task perspective divides methods by different summarization tasks.
- **SDS versus MDS:** Based on the number of documents, summarization methods are divided into SDS or MDS methods.
- **Query-Based versus Generic Summarization:** If a user-specific information type is required, a query-based summarization is provided. While in generic summarization, all relevant topics should be included in the summary.
- **Monolingual/Multilingual versus Cross-lingual Summarization:** Based on language coverage, systems are divided into three categories: monolingual, multilingual, and cross-lingual. In monolingual and multilingual summarizing systems, the language of the input documents and the summary are the same. The first one deals with only one language, but the latter works in several languages. In contrast, cross-lingual systems can process input documents in different languages, as well as summarize them in different languages.
- **Headline Generation:** In the task of headline generation, the purpose is generating a very short summary as a headline for a text.
- **Question Answering:** Question answering systems deal with a question and a document (or a set of documents) which seems to be relevant to the question. The task is to create a short summary of the document answering that question.

2-1- Definition of Multi-document Summarization

MDS is a way to display the main content of a set of documents with a similar topic by a short text by including important and relevant information and filtering out the redundant information. Two prominent approaches in summarizing multiple documents are extractive and abstractive summarization. Extractive systems aim to extract prominent sentences from documents, while abstractive summarization systems aim to paraphrase the contents of documents to generate a new shortened text [9].

2-2- Applications of Multi-document Summarizers

Text summarization has many applications in today's world. For example, the production of a summary of various E-books or scientific articles, the production of summaries of patients' medical information, generation a site summary (what search engines like Google do), summarizing product reviews, student responses to classroom questionnaires, and a series of news articles about a specific topic [10].

Of course, these are only a few examples of the many uses of this topic in today's society.

2-3- Categorizations on MDS

Some researchers have provided categorizations for MDS summarizing methods. Joshi and Kadam have divided MDS into three categories: cluster-based approaches, ranking-based approaches, and LDA-based approaches [5]. Tabassum and Oliveria have divided MDS into five types from another perspective: feature-based approaches, domain-specific (ontology-based) approaches, cognitive-based approaches, event-based approaches, and discourse-based approaches [11].

Shah and Jivani have done another categorization on MDS methods [12]. From their point of view, MDS methods can be classified into four categories: graph-based approaches, cluster-based approaches, term frequency-based methods, and Latent Semantic Analysis (LSA).

In another study, Tandel et al. classified MDS methods into three categories: cluster-based approaches, topic-based approaches, and lexical chain approaches [13].

Gupta and Lehal categorized summarization methods into these categories:

Various categorizations on MDS presented by different researchers [5], [11], [12], and [13] are shown in Fig 2.

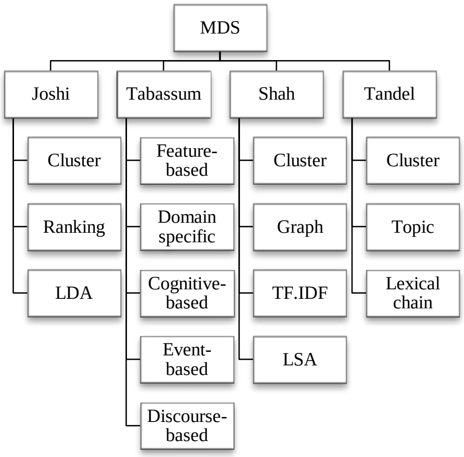


Fig. 2 Various categorizations on MDS Source: [5], [11], [12], and [13]

3- Proposed Categorization on MDS

We have also proposed a more comprehensive categorization for MDS methods by reviewing the previous works in this field. The Categorization presented in this paper for MDS methods is shown in Fig 3. We categorize MDS methods into six categories: machine learning, clustering, graph, LDA, optimization, and deep learning.

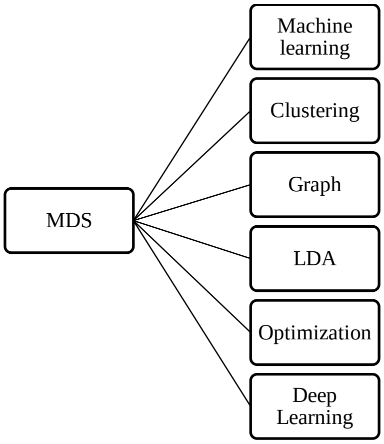


Fig. 3 Categorization of MDS approaches

3-1- Machine Learning-based Approaches:

Machine learning methods are widely used in the summarization process. For example SVM [14], ANN [15], and Decision Tree [16] are various types of machine learning methods used in summarizing.

One of the SVM-based MDS techniques was proposed by Chali et al [14]. They used an ensemble of SVMs. Each SVM made its own prediction for an unseen sentence. The SVM ensemble then combined these predictions based on the weighted averaging technique and produced the final prediction. Four different SVMs were used to rank the sentences, and finally the top-ranked sentences were selected for the final summary.

Neural networks have also been used to summarize multiple documents. One such study was conducted in 2016 by Ma et al. [15]. They proposed an unsupervised multi-document summarization framework based on a neural document model. In this method, a neural document model in the multi-document summarization task and a document level reconstruction framework called DocRebuild are proposed. Neural document model tries to display the semantic content of documents using low-dimensional vector representations. In this method, two types of unsupervised neural document models called Bag-of-Words (BOW) and Paragraph Vector (PV) are used to represent the semantic content of documents. DocRebuild reconstructs documents with summary sentences through a neural document model and selects summary sentences to minimize reconstruction errors.

The decision tree has also been used to summarize multiple documents. Ou et al. used decision trees to design their multi-document summarization system [17]. They proposed a multi-document summarization system for sociology dissertation abstracts. Their proposed system performs a discourse parsing after receiving and pre-processing a set of related dissertation abstracts on a specific topic. In this step, using a decision tree classifier, each sentence is classified into one of five predefined sections: *background*, *research objectives*, *research methods*, *research results* and *concluding remarks*. Then in the information extraction stage, important concepts are extracted from each of the sections using pattern matching. Then, in the information integration stage, the obtained information is clustered so that similar concepts are in the same clusters. Finally, based on this integrated information, a final summary is generated.

3-2- Clustering-based Approaches

Clustering-based methods have also been used to summarize the text. These methods are able to identify various topics raised in the texts. One of the works in this field was done by Gupta and Siddiqui [18]. Their method was multi-document and query-based.

This method first generated an SDS for each text. All SDSs were then combined and a clustering was applied to the set of all sentences. In the clustering process, syntactic and semantic similarities between sentences were considered. From each cluster, the most important sentence was selected, and the set of selected sentences was sorted based on their location in the main document and constitute the final summary.

Cai and Li proposed an integrated approach using ranking through clustering for MDS [19]. Applying clustering to documents leads to the production of a number of topic themes. Each theme is displayed by a cluster of highly related sentences. For each topic theme, the term ranks in that topic theme should be different and distinct from term ranks in other topic themes. Most existing cluster-based summarization systems apply clustering and ranking methods individually, which results in incomplete or sometimes biased results. In Cai and Li’s approach clustering results are used to improve or refine the sentence ranking results. The main idea of this method is that the ranking distribution of sentences in each cluster should be different. In their approach ranking and clustering simultaneously update each other, and the performance of both is improved.

3-3- Graph-based Approaches

Graph-based methods for MDS are widely used to extract top sentences for summaries. Al-Dhelaan presents a simple star graph for MDS called StarSum [20]. StarSum is a star bipartite graph that models sentences and their topic signature phrases. This method extracts sentences that guarantee the diversity and coverage, both of which are essential for MDS. The diversity is guaranteed by splitting the StarSum graph into different components and using top sentences from each of the different components. Coverage is guaranteed by ranking sentences by their degree of connection to other topic sentences and phrases.

Khan et al. present a clustered semantic graph approach for multi-document abstractive summarization [21]. Most of the existing graph-based methods rely on the bag of words method, which treats sentences as a bag of words and relies on a content similarity measure. The main limitation of the bag of words method is that it does not consider the relationship between words. This paper proposes a clustered semantic graph-based method for abstractive MDS. This method uses Semantic Role Labeling (SRL) to extract the semantic structures called Predicate Argument Structure (PAS). Pairwise PASs are compared based on the linear semantic similarity measure to create a semantic similarity matrix. This matrix is represented as a semantic graph. PASs are vertices of the graph. The edges represent the semantic similarity weight between the vertices. For content selection, graph nodes (PASs) are ranked based on the modified graph-based ranking algorithm. Maximal Marginal Relevance (MMR) is applied to redundancy reduction. PAS representatives with the highest salience score are selected from each cluster and are fed to language generation to generate summary sentences.

Glavas and Snajder proposed event graphs for MDS [22]. They first created an event-based document representation of the text. This representation is an event graph that contains information about the events described in the text. Event mentions are then extracted from the sentences. Then event graph information is used to understand the importance of the events in the sentence and the relationships between the events. Each sentence gets a score according to the importance of the events presented in it and is selected according to these scores to attend the summary.

3-4- LDA-based Approaches

LDA dates back to 2003 [23]. It is a generative probabilistic model. LDA in natural language processing can do topic discovery. The basic idea of the LDA is that documents are displayed as random mixtures on latent topics, where each topic is a distribution of words. A set of documents has a probabilistic distribution of topics so that a particular document probably contains some topics more than others. Terms within a topic also have their own probability distribution. That is, in a topic, some specific terms are used much more than others. In LDA, both sets of probabilities in the training phase are calculated using the Bayesian methods and the expectation maximization algorithm. Some researchers have used LDA to perform MDS. LDA-based extractive summarization methods, after discovering the hidden topics in the text, select the most important sentences in each topic as a representative of that topic to participate in the summary.

Roul [24] proposed a topic-based model for extractive multi-document text summarization. The model identifies the number of independent topics using LDA and three probabilistic models. The probabilistic methods Hellinger’s distance, Jensen Shannon Divergence, and KL divergence are used to compute the similarity between each of the topic pairs. Then LDA technique is used again to reduce a large set of n sentences to a smaller set while maintaining the important information. The representative sentence from each topic is selected and arranged by the corresponding topic importance to appear in the summary.

One of the other researches in this field has been done by Na et al [25]. They proposed a method called Titled-LDA. Titled-LDA is an extended version of LDA that does both title topic modeling and content topic modeling. The two models are then combined to create a new mixture topic model. In the mixing step, a weight is assigned to each of the two distributions. These weights are learned by an adaptive asymmetric learning algorithm. Titled-LDA consists of five steps: (1) Extracting title and content from each document. (2) Obtaining title and content topic distributions. (3) Combining topic models using adaptive asymmetric learning algorithm. (4) Calculating sentence scores based on mixture topic model. (5) Generate summaries based on sentence scores.

3-5- Optimization-based Approaches

The MDS task faces challenges such as redundancy, complementarity, and contradiction [26]. To generate informative extractive MDS, the most important set of sentences should be selected, avoiding redundancy and contradiction, and maintaining complementarity between them. Each of the phenomena can be considered as an objective function and optimization methods can be used to solve it.

Su et al. [27] proposed an optimization-based MDS called PoBRL (Policy Blending with maximum marginal relevance and reinforcement Learning). PoBRL considers three objectives: importance, non-redundancy and length for the summarization task. This multi-objective optimization task is broken down into different sub-tasks, each of which is solved separately by reinforcement learning. The learned policies are then combined by PoBRL to produce the final summary.

Alguliev et al. consider MDS as an evolutionary optimization problem [28]. The sentence-to-document collection, summary-to-document collection and sentence-to-sentence relationships are used to select salient sentences from the document collection and reduce redundancy. They consider this problem as a discrete optimization problem. To solve the discrete optimization, a self-adaptive Differential Evolution (DE) algorithm has been created.

Sanchez-Gomez et al. provided an extractive MDS using a multi-objective artificial bee colony optimization approach [29]. The extractive MDS methods aim to obtain the main content of a data collection and simultaneously reduce redundant information. The paper analyzes this issue from the perspective of optimization. For this purpose, a Multi-Objective Artificial Bee Colony (MOABC) algorithm is proposed. The MDS problem needs to be optimized for more than one objective function and so Multi-Objective Optimization can be used for this purpose. MOABC has three types of bees that allow different search mechanisms for any bee. Employed bees maintain current solutions. Onlooker bees allow exploitation of the best solutions ever found. Scout bees eliminate stagnated solutions and allow exploration of the partially good solutions. This combination of exploration and exploitation mechanisms provides an effective way for MDS.

John et al. formulated extractive MDS as population-based multi-criteria optimization [30]. They consider three objective functions for determining an optimal summary: maximum relevance, diversity, and novelty. For this purpose, both syntactic and semantic aspects of the document are considered. The semantic aspect is considered through LSA techniques and Negative Matrix Factorization. In each iteration of the algorithm, three candidate summaries are identified that maximize the value of the objective functions and create the final optimal summary. Table 1 shows comparison matrix of machine learning, clustering, graph, LDA, and optimization approaches.

3-6- Deep Learning-based Approaches

In recent years, deep learning has gained significant results in many areas of NLP, including text summarization. Many scholars have focused their attention on deep learning methods for MDS.

Afsharizadeh et al. [31] proposed an extractive summarization using Recurrent Neural Networks (RNN) and coreference resolution procedure. The model stores coreference information in the form of coreference vectors. A three-layer Bidirectional Long Short Term Memory (Bi-LSTM) computes sentence representations using the embedding vectors of their constituent words. The sentence representations are then enriched using the coreference vectors.

Zhang et al. provide a multiview convolutional neural network for extractive MDS [32]. In this paper, an extended CNN was used to obtain sentence features and to rank sentences. Multiview learning was added to the model to improve CNN's ability to learn. Three CNN networks are used to generate a summary. Each CNN has a salience score for each sentence. Then these scores are combined to get the final score for the sentences.

Cao et al. develop a ranking framework for the recursive neural network to rank sentences in the MDS [33]. This article formulated the sentence ranking task as a hierarchical regression process. Recursive neural networks are used to learn auto-ranking features: learned features supplement hand-crafted features to rank sentences. Finally, ranking score sentences are used to effectively select informative and non-redundant sentences. Zhong et al. present a query-oriented unsupervised MDS through a deep learning model [34].

Table 1: Comparison matrix of machine learning, clustering, graph, LDA, and optimization approaches

Work	Category	Purpose	Dataset	Results
A SVM-based ensemble approach to multi-document summarization [14]	Machine learning: SVM	Using an ensemble of SVMs for MDS.	DUC 2007	**R-1:** 0.388 **R-2:** - **R-L:** 0.319 **R-SU:** 0.146
Unsupervised multi-document summarization framework based on neural document model [15]	Machine learning: NN	Using a document level reconstruction framework using neural document model for MDS.	DUC 2006 DUC 2007	**R-1:** 0.421, 0.434 **R-2:** 0.093, 0.105 **R-L:** - **R-SU:** 0.151, 0.162
Summarizing Indonesian text automatically using sentence scoring and decision tree [17]	Machine learning: Decision tree	Using a decision tree to classify each sentence into one of five predefined categories. Then extracting main concepts using pattern matching.	50 text documents	**R-1:** 0.580 **R-2:** - **R-L:** - **R-SU:** -
Multi-document summarization using sentence clustering [18]	Clustering	Applying clustering to a set of SDSs to make a multi-document summary.	DUC2002	**R-1:** 0.338 **R-2:** - **R-L:** - **R-SU:** -
Ranking through clustering: An integrated approach to multi-document summarization [19]	Clustering	Using ranking through clustering for MDS	DUC2004 DUC2005 DUC2006 DUC2007	**R-1:** 0.374, 0.364, 0.405, 0.416 **R-2:** 0.089, 0.073, 0.093, 0.120 **R-L:** - **R-SU:** -
StarSum: A Simple Star Graph for Multi-document Summarization [20]	Graph	Using a star bipartite graph that models sentences and their topic phrases to summarize multiple documents.	DUC2001	**R-1:** 0.523 **R-2:** 0.391 **R-L:** 0.511 **R-SU:** -
A clustered semantic graph	Graph	Using SRL to extract the semantic structures of the text	DUC2002	**R-1:** 0.400

Work	Category	Purpose	Dataset	Results
approach for multi-document abstractive summarization [21]		and represents them as a semantic graph. Nodes with the highest salience score are selected from each cluster generate summary sentences.		**R-2:** 0.099 **R-L:** - **R-SU:** -
Event graphs for information retrieval and multi-document summarization [22]	Graph	Proposing event graphs that contain information about the events described in the text for MDS.	DUC2002 DUC2004	**R-1:** 0.415, 0.405 **R-2:** 0.116, 0.107 **R-L:** - **R-SU:** -
Topic modeling combined with classification technique for extractive multi-document text summarization [24]	LDA	Using LDA and probabilistic models for text topic identification. Sentences with highest topic importance scores are selected for the summary.	DUC2002 DUC2006	**R-1:** 0.497, 0.429 **R-2:** 0.258, 0.094 **R-L:** - **R-SU:** -
Mixture of topic model for multi-document summarization [25]	LDA	Mixing title topic modelling and content topic modelling to create a new mixture topic model and using it for MDS.	DUC2002	**R-1:** 0.463 **R-2:** 0.182 **R-L:** 0.422 **R-SU:** 0.226
PoBRL: Optimizing Multi-document Summarization by Blending Reinforcement Learning Policies [27]	Optimization	Using a multi-objective optimization-based approach. Each objective is solved separately by reinforcement learning. The learned policies are then combined to produce the final summary.	MultiNews DUC2004	**R-1:** 0.465, 0.386 **R-2:** 0.173, 0.102 **R-L:** 0.424, 0.131 **R-SU:** -
Multiple documents summarization based on evolutionary optimization algorithm [28]	Optimization	Considering MDS as a discrete optimization problem. A self-adaptive DE algorithm is used to solve it.	DUC2002 DUC2004	**R-1:** 0.499, 0.393 **R-2:** 0.258, 0.112 **R-L:** 0.489, 0.396 **R-SU:** 0.287, 0.135
Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach [29]	Optimization	Proposing a multi-objective artificial bee colony optimization approach. The model has three types of bees. The combination of them provides an effective way for MDS.	DUC2002	**R-1:** - **R-2:** 0.312 **R-L:** 0.540 **R-SU:** -
Extractive multi-document summarization using population-based multicriteria optimization [30]	Optimization	Formulating extractive MDS as population-based multi-criteria optimization. Three objectives are used to consider both syntactic and semantic aspects of the text.	DUC2002 DUC2004 DUC2006	**R-1:** 0.548, 0.521, 0.325 **R-2:** 0.271, 0.171, 0.069 **R-L:** - **R-SU:** -

The proposed framework includes three parts: concept extraction, summary generation, and reconstruction validation. The deep Auto Encoder (AE) network is used for this purpose. The concept extraction part is the phase of encoding of the network and obtaining a compact representation of the concept. The reconstruction validation phase relates to the network decoding phase and attempts to reconstruct the inputs of the network. The summary generation step using dynamic programming generates a final summary of the candidate sentences.

Lakshmi and Rani provide a method for implementing MDS using deep learning and fuzzy logic [35]. Restricted Boltzmann Machines (RBM) are used to produce a shortened version of the document without losing its important information. First, the text is converted to a feature matrix, in which the rows are related to sentences and columns to features. The fuzzy classifier then assigns labels to sentences. A new feature matrix is formed by adding a column of labels to sentences. This feature matrix is considered as the input for the RBM which receives input from each row of this feature matrix. The RBM learns network weights by trying to reconstruct inputs. After generating a score for sentences, high ranked sentences are selected to be in the summary.

Table 2 shows a brief summary of deep learning-based approaches. Also, advantages and disadvantages of the methods used in MDS are shown in Table 3.

4- Domain-oriented Summarization

Sometimes a text is related to a specific domain. In this case, it often has a specific structure or characteristics unique to its domain type. Such characteristics help summarization algorithms to more accurately identify the most important information and provide a more detailed summary. For example, journal articles often have an abstract and conclusion section that contains the most important information about the text [36]. Multi-document summarizers have been applied in a wide range of domains, such as summarizing scientific articles, literary texts, blog posts, and patient data [37]. Accordingly, a multi-document summarizer can be a domain-oriented or domain-independent approach.

Some MDS systems are specifically designed for a particular genre of documents, for example, news articles about terrorism [38]. SUMMONS (SUMMarizing Online NewS Articles) was proposed as a summarizer of news articles [39]. The input of this system is a collection of generated templates by the MUC (Message Understanding Conference), which works on the domain of terrorism. Each template shows the information extracted from one or more articles. The templates are then compared and merged using different planning operators. Each operator combines a pair of templates for a new template. There are seven operators in SUMMONS including agreement, addition, contradiction, etc

Table 2: Comparison of some deep learning-based approaches

Paper Title	Purpose	Dataset	Results
Automatic Text Summarization of COVID-19 Research Articles Using Recurrent Neural Networks and Coreference Resolution [31]	Using a combination of RNNs and coreference resolution procedure for summarization. The model stores coreference information in the form of coreference vectors.	CORD19	**R-1:** 0.343 **R-2:** 0.116 **R-L:** 0.188 **R-SU:** 0.152
Multiview convolutional neural networks for multi-document extractive summarization [32]	Provide a multiview convolutional neural network for extractive MDS. Multiview learning was added to the model to improve CNN's ability to learn.	DUC2001 DUC2002 DUC2004 DUC2006 DUC2007	**R-1:** 0.359, 0.367, 0.390, 0.386, 0.409 **R-2:** 0.079, 0.090, 0.100, 0.079, 0.091 **R-L:** - **R-SU:** 0.131, 0.149, 0.136, 0.140, 0.153
Ranking with Recursive Neural Networks and Its Application to Multi-document Summarization [33]	Developing a ranking framework for the recursive neural network to rank sentences in the MDS.	DUC2001 DUC2002 DUC2004	**R-1:** 0.369, 0.379, 0.387 **R-2:** 0.078, 0.088, 0.098 **R-L:** - **R-SU:** -
Query-oriented unsupervised multi-document summarization via deep learning model [34]	Presenting a query-oriented unsupervised MDS through a deep learning model. The deep AE network is used for this purpose.	DUC2005 DUC2006 DUC2007	**R-1:** 0.375, 0.401, 0.429 **R-2:** 0.077, 0.092, 0.116 **R-L:** - **R-SU:** 0.134, 0.147, 0.168
Multi-document Text Summarization Using Deep Learning Algorithm with Fuzzy Logic [35]	Implementing MDS using RBM and fuzzy logic.	DUC2002	**R-1:** 0.550 **R-2:** - **R-L:** - **R-SU:** -

Table 3: Advantages and disadvantages of different approaches in the field of MDS

Method	Advantage	Disadvantage
Machine Learning	Using different Machine Learning methods such as Bayes Classifiers, Artificial Neural Networks, and SVM	**The main drawback of this method is obtaining a labeled dataset. Labeling sentences in documents is a time-consuming operation.**
Clustering	Clustering-based methods are suitable for texts with multiple different topics.	**K-means is the most famous clustering algorithm. One of its disadvantages is determining the appropriate value for k, i.e. the number of clusters. This method is also suitable for clusters with spherical shapes but not for non-convex shapes.**
Graph	These methods create a good visual representation of the text, which at a glance can identify the number of distinct topics in the text and the most important sentence in each topic.	**Choosing a way to score vertices is challenging.**
LDA	Ability to discover hidden topics in the text.	**The number of topics is fixed and must be determined in advance.**
Optimization	Sentences contained in the multi-document summary should be relevant, non-redundant, and non-contradict. They can be considered as objective functions. Then MDS is solved as an optimization problem. The goal is to find the best settings for summarization.	**This method may be slow since finding the optimal weights for objective functions is achieved after several iterations.**
Deep Learning	Ability to auto-learn features from raw data, suitable for large datasets	**These methods require a large amount of training data to learn model parameters.** **Possibility of vanishing and exploding gradient problems that cause the model not to be trained and therefore incorrect adjustment of parameters**

SUMMONS architecture consists of two main components: content planner and linguistic generator. Content planner generates a conceptual representation of the meaning of the text, and usually does not contain any linguistic information. The content planner determines what information in input templates should be included in the summary, and does this with a set of planning operators. An operator is used to link information in two different templates. A summary can be the result of applying a single operator. More complex summaries can be generated by multiple operators. The linguistic component consists of lexical chooser and sentence generator components.

The lexical chooser defines a high level sentence structure for each sentence. The sentence generator, using a large English grammar, meets the syntactic constraints, creates a syntactic tree, and linearizes the tree as a sentence. Radev suggested a Cross-Document Structure Theory (CST) that was a taxonomy of the relationship between the documents [40].

The CST concept is similar to the discourse structure in a single document. These cross-document relationships can be used in MDS, and some of them are a direct descendant of the ones used in SUMMONS.

When the input of the summarization system is related to a particular domain, conventional summarization methods may not be appropriate. When input documents include specialized information in a particular domain, they often have a specific structure and characteristics. These characteristics can help summarization algorithms to more accurately identify important information [36]. For example, journal articles often have a conclusion section that includes the key information of the article and contains important information for the summary. Certain domains like medical or law may have specific requirements for the type of information that is required in a summary. Such domains may also have resources that can help the summarization process. In the following, a number of summarization methods are reviewed in the medical domain.

Conventional summarization methods are not easily applicable to certain domains, such as the medical domain. In this domain, summarizing algorithms, with the precise use of specific medical definitions, have valuable applications such as helping clinicians in their treatment cycle, reviewing the latest research on a particular patient, or helping patients and their families with information about the disease. Medical articles have a specific structure that algorithms take. In addition, there are also extensive knowledge resources available in the medical field. The final users of medical summarizing systems are healthcare providers and consumers, who both can access information of interest through the Internet. In the medical community, the number of journals related to even a single field is very high and it is difficult for physicians to know about all the new results reported in their specialized fields. Similarly, patients and their family members who need information about their particular illness face a huge amount of online information, which ultimately leads to more confusion. The summary can be designed based on the type of user that is a healthcare provider or a patient. There are important sources of information on healthcare. An ontology of medical concepts, the Unified Medical Language System (UMLS), is available and can be automatically linked to terms in input articles [41].

The input for each concept can be done in several ways. Centrifuser is a summarizer that aims to help better search for information [42]. Centrifuser is an extractive domain oriented query-based MDS. The produced summary has three parts: (1) Link to query related topics for easier navigation and query reformulation. (2) A high-level overview of common parts of documents. (3) A description of the difference between recovered documents to guide people to select related items.

Medical literature on the Web is an important resource for clinicians to care for the patient [43]. A summary of medical contents will help clinicians and medical students find important and relevant information on the web faster. The presented summarization method combines various domain-specific features with other well-known features such as frequency, title, and position to improve summarization performance in the medical domain. This summarizer consists of three parts: document preprocessing, sentence ranking and summary generation. In the document preprocessing process, actions such as sentence segmentation, tokenization, stemming, and stop words removal are done. In the sentence ranking phase, some scores should be considered for the sentences. This score is calculated in terms of several factors: 1- Term frequencies. 2- Sentence similarity to the document’s title. 3- Position of the sentence in the text. 4- The presence of domain related terms in the sentence. 5- The presence of new terms in the sentence. 6- Summary length.

In the summary generation phase, sentences with higher scores are selected for the summary.

Sentence ranking phase uses a knowledge base for identifying domain related terms in each sentence. This knowledge base is a list of medical terms and phrases with their weights that has been prepared by a corpus of medical news articles. In preparing the knowledge base, at first cue phrases receive a weight between 1 and 8 in terms of their impact on determining the summary worthiness and then each weight adds additional values in the range of 0 to 2 based on their position in the sentences. The position of a cue phrase in the sentence is important. A sentence that contains a cue phrase at the beginning receives a higher score than in the other positions. So a cue phrase receives a weight between 1 and 10. The knowledge base is required to identify the medical cue terms and phrases in the sentences (step (4) in sentence ranking phase). However new medical terms, such as names of genes, drugs, and diseases, are also discovered at any time. For this reason, the article introduces the idea to identify new medical terms in the text (step (5) in sentence ranking phase). An algorithm for novel medical term detection is used for this purpose. The algorithm uses two different vocabularies to determine if a term is a new medical term. One of them is a medical vocabulary and the other is an English vocabulary. If a word does not appear in the medical vocabulary, it should not be considered as a new word, because medical articles are also written using natural language, which includes words such as verbs, adverbs, and some nouns that may not be present in the medical vocabulary. Therefore, a different vocabulary is used that is made up of a corpus of natural language texts (not the medical domain). If a word was not in any of these two vocabularies, it would be considered as a novel medical term.

5- Datasets and Evaluation

5-1- Datasets

Researchers have used various datasets in the text summarization task. Many of the analyses in MDS are done on the Document Understanding Conference (DUC) datasets. This conference was organized by the National Institute of Technology (NIST) [44]. DUC has become the main international forum for discussion on text summarization [8]. The conference was first held in 2001. In each DUC edition, one or more specific tasks have been reviewed. DUC 2001 and DUC 2002 had two tasks, SDS and MDS.

Four tasks were defined for DUC 2003: 1- headline generation, which aims to produce a short 10-words summary from the single document. 2- MDS to generate summaries focusing on events in the text. 3- MDS to generate summaries focusing on viewpoints. A viewpoint is a natural language string whose length is slightly larger than a sentence and describes the important facets of the cluster. 4- MDS aims to produce short summaries of each cluster to answer a question that comes with each cluster. The requested task in DUC 2004 is to produce cross-lingual single / MDS for English and Arabic. DUC 2005 and DUC 2006 combine a set of documents to produce a brief, well-organized, fluent answer to the question. Two tasks are defined for DUC 2007. The first task is question answering, and the second task is to generate a multi-document update summary from newswire articles, assuming that the user already knows basic information about the topic. The purpose of any update summary is to give the reader new information about a specific topic. The general characteristics of the DUC data are shown in Table 4.

5-2- Evaluation

The most popular measure for evaluating summarization methods, including MDS, is ROUGE [45]. ROUGE stands for Recall Oriented Understudy for Gisting Evaluation. It measures the overlap of textual units between the system summary and the reference summary (or set of reference summaries). ROUGE evaluates the system summary from two different perspectives: recall and precision. Recall-oriented ROUGE, Precision-oriented ROUGE and F-measure-oriented ROUGE are computed by the equations (1) to (3), respectively.

$$R_r = \frac{|overlapping\ n-grams|}{|n-grams\ in\ Ref\ summary|} \quad (1)$$

$$R_p = \frac{|overlapping\ n-grams|}{|n-grams\ in\ Sys\ summary|} \quad (2)$$

$$R_f = \frac{2 \times R_p \times R_r}{R_p + R_r} \quad (3)$$

Rr: What percentage of the reference summary is covered by the system summary? This is a sign of perfection.

Rp: What percentage of the system summary is covered by the reference summary? This shows usefulness.

It is desirable that both be high, which leads to an increase in Rf. ROUGE includes a family of measures: ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, and ROUGE-SU. ROUGE-N works on n-grams. It measures the overlap of unigrams between system and reference summaries. ROUGE-N is calculated using the equation (4):

$$ROUGE - N = \quad (4)$$
$$= \frac{\sum_{S\in\{Ref\ Sum\}} \sum_{gram_n \in S} Count_{match}\,(gram_n)}{\sum_{S\in\{Ref\ Sum\}} \sum_{gram_n \in S} Count\,(gram_n)}$$

ROUGE-L works with the concept of LCS. LCS is the Longest Common Substring. It calculates the overlap of LCS units between system and reference summaries. One drawback of ROUGE-L is that it does not differentiate between substrings with different distances between their elements. ROUGE-W considers the length of consecutive matches as a weight. skip-gram is any pair of words in the text with any number of gaps between them. ROUGE-S considers skip-garm overlaps between system and reference summaries. One drawback of ROUGE-S is that it does not score to a system summary that has no common skip-grams with the reference summary. That's why ROUGE-SU has combined ROUGE-S with unigrams. Between several ROUGE evaluation measures, ROUGE-1, ROUGE-2, ROUGE-Sn, and ROUGE-SUn are usually used to evaluate MDS summaries [46].

6- Challenges and Recommendations

Multi-document summarizers face more challenges than single document summarizers. For example, redundancy occurs more often in summarizing multiple documents related to a similar topic than in summarizing a single document. On the other hand, MDS performs summarization at a higher compression ratio than SDS. This makes summarizing operations more difficult. Also, summaries generated from multiple documents may be less readable than summaries generated from a single document. This is because the collection of sentences, each of which belongs to one of the documents, may result in poor coherence of the generated summary text.

There are some recommendations for each of these challenges. For example, for the information redundancy problem, before adding any candidate sentence to the summary, it is possible to calculate its overlap with the partial summary that has been produced so far. If there is a high amount of overlap, that sentence does not add to the summary. It is also possible to increase the length constraint for summaries produced from multiple documents. So there is no need to summarize at a high compression ratio. The problem of non-readability of summaries generated from multiple documents can also be reduced with techniques such as coreference resolution.

Table 4: General characteristics of DUC data [8]

Year	**DUC 2001**	**DUC 2002**	**DUC 2003**	**DUC 2004**	**DUC 2005**	**DUC 2006**	**DUC 2007**
Number of input document sets	30	59	30	50	50	50	**45**
Number of documents per set	6-16	5-15	10	10	25-50	25	**25**
Number of human summaries	3	2	4	4	4-9	4	**4**
Summary length	50/100/200/400	50/100/200	100	100	250	250	**250**

7- Conclusion

Searching the web for a specific topic leads to retrieving hundreds of related documents. Multi-document summarization allows the user to access the most important content of multiple text documents in a short time. In this survey we have attempted to give a comprehensive overview of the multi-document summarization techniques and domain-oriented approaches. We have categorized multi-document summarization techniques into six groups: machine learning, clustering, graph, LDA, optimization, and deep learning. A comparative overview of recent developments in each of the categories is provided. We have expressed the strengths and weaknesses of each category. Sometimes it is better to consider the domain type of the input text document, for example documents related to medicine, law, geography, etc. In this article, we also have investigated domain-oriented techniques. We have also described the most famous datasets and evaluation measures in the text summarization studies. Finally, a number of challenges and recommendations have been presented.

References

[1] G. Carenini and J. C. K. Cheung, "Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality," in Proceedings of the Fifth International Natural Language Generation Conference, 2008, pp. 33-41.

[2] A. Abdi, N. Idris, R. M. Alguliyev, and R. M. Aliguliyev, "Query-based multi-documents summarization using linguistic knowledge and content word expansion," Soft Computing, vol. 21, pp. 1785-1801, 2017.

[3] C. Ma, W. E. Zhang, M. Guo, H. Wang, and Q. Z. Sheng, "Multi-document Summarization via Deep Learning Techniques: A Survey," arXiv preprint arXiv:2011.04843, 2020.

[4] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz, "Multi-document summarization by sentence extraction," in Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization, 2000, pp. 40-48.

[5] R. R. K. Parchi M. Joshi, "Survey on Multi-document Summarizer," International Journal of Science and Research (IJSR), vol. 3, p. 5, 2014 2014.

[6] N. Andhale and L. Bewoor, "An overview of text summarization techniques," in Computing Communication Control and automation (ICCUBEA), 2016 International Conference on, 2016, pp. 1-7.

[7] M. Yousefiazar, "Query-oriented single-document summarization using unsupervised deep learning," 2015.

[8] M. Fuentes Fort, A flexible multitask summarizer for documents from different media, domain and language: Universitat Politècnica de Catalunya, 2008.

[9] K. Mani, I. Verma, H. Meisheri, and L. Dey, "Multi-document summarization using distributed bag-of-words model," in 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2018, pp. 672-675.

[10] L. Lebanoff, K. Song, and F. Liu, "Adapting the Neural Encoder-Decoder Framework from Single to Multi-document Summarization," arXiv preprint arXiv:1808.06218, 2018.

[11] S. Tabassum and E. Oliveira, "A review of recent progress in multi-document summarization," in Doctoral Symposium in Informatics Engineering, 2015.

[12] C. Shah and A. Jivani, "Literature study on multi-document text summarization techniques," in International Conference on Smart Trends for Information Technology and Computer Communications, 2016, pp. 442-451.

[13] A. Tandel, B. Modi, P. Gupta, S. Wagle, and S. Khedkar, "Multi-document text summarization-a survey," in Data Mining and Advanced Computing (SAPIENCE), International Conference on, 2016, pp. 331-334.

[14] Y. Chali, S. A. Hasan, and S. R. Joty, "A SVM-based ensemble approach to multi-document summarization," in Canadian Conference on Artificial Intelligence, 2009, pp. 199-202.

[15] S. Ma, Z.-H. Deng, and Y. Yang, "An unsupervised multi-document summarization framework based on neural document model," in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 1514-1523.

[16] P. M. Sabuna and D. B. Setyohadi, "Summarizing Indonesian text automatically by using sentence scoring and decision tree," in Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2017 2nd International conferences on, 2017, pp. 1-6.

[17] S. Ou, C. S. Khoo, and D. H. Goh, "A multi-document summarization system for sociology dissertation abstracts: design, implementation and evaluation," in International Conference on Theory and Practice of Digital Libraries, 2005, pp. 450-461.

[18] V. K. Gupta and T. J. Siddiqui, "Multi-document summarization using sentence clustering," in Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on, 2012, pp. 1-5.

[19] X. Cai and W. Li, "Ranking through clustering: An integrated approach to multi-document summarization," IEEE transactions on audio, speech, and language processing, vol. 21, pp. 1424-1433, 2013.

[20] M. Al-Dhelaan, "StarSum: A Simple Star Graph for Multi-document Summarization," in Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 715-718.

[21] A. Khan, N. Salim, W. Reafee, A. Sukprasert, and Y. J. Kumar, "A clustered semantic graph approach for multi-

document abstractive summarization," Jurnal Teknologi (Sciences & Engineering), vol. 77, pp. 61-72, 2015.

[22] G. Glavaš and J. Šnajder, "Event graphs for information retrieval and multi-document summarization," Expert systems with applications, vol. 41, pp. 6904-6916, 2014.

[23] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," the Journal of machine Learning research, vol. 3, pp. 993-1022, 2003.

[24] R. K. Roul, "Topic modeling combined with classification technique for extractive multi-document text summarization," Soft Computing, vol. 25, pp. 1113-1127, 2021.

[25] L. Na, L. Ming-xia, L. Ying, T. Xiao-jun, W. Hai-wen, and X. Peng, "Mixture of topic model for multi-document summarization," in Control and Decision Conference (2014 CCDC), The 26th Chinese, 2014, pp. 5168-5172.

[26]J. W. da Cruz Souza and A. Di Felippo, "Characterization of Temporal Complementary: Fundamentals for Multi-Document Summarization /Caracterizacao da complementaridade temporal: subsidios para sumarizacao automatica multidocumento," Alfa: Revista de Lingüística, vol. 62, pp. 121-148, 2018.

[27] A. Su, D. Su, J. M. Mulvey, and H. V. Poor, "PoBRL: Optimizing Multi-document Summarization by Blending Reinforcement Learning Policies," arXiv preprint arXiv:2105.08244, 2021.

[28] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "Multiple documents summarization based on evolutionary optimization algorithm," Expert Systems with Applications, vol. 40, pp. 1675-1689, 2013.

[29]J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach," Knowledge-Based Systems, vol. 159, pp. 1-8, 2018.

[30] A. John, P. Premjith, and M. Wilscy, "Extractive multi-document summarization using population-based multicriteria optimization," Expert Systems with Applications, vol. 86, pp. 385-397, 2017.

[31] M. Afsharizadeh, H. Ebrahimpour-Komleh, and A. Bagheri, "Automatic Text Summarization of COVID-19 Research Articles Using Recurrent Neural Networks and Coreference Resolution," Frontiers in Biomedical Technologies, vol. 7, pp. 236-248, 2020.

[32] Y. Zhang, M. J. Er, R. Zhao, and M. Pratama, "Multiview convolutional neural networks for multidocument extractive summarization," IEEE transactions on cybernetics, vol. 47, pp. 3230-3242, 2017.

[33] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, "Ranking with Recursive Neural Networks and Its Application to Multi-document Summarization," in AAAI, 2015, pp. 2153-2159.

[34] S.-h. Zhong, Y. Liu, B. Li, and J. Long, "Query-oriented unsupervised multi-document summarization via deep learning model," Expert Systems with Applications, vol. 42, pp. 8146-8155, 2015.

[35] S. S. Lakshmi and M. U. Rani, "Multi-document Text Summarization Using Deep Learning Algorithm with Fuzzy Logic," 2018.

[36] A. Nenkova and K. McKeown, "Automatic summarization," Foundations and Trends® in Information Retrieval, vol. 5, pp. 103-233, 2011.

[37] S. Kasundra and D. L. Kotak, "Study on Multi-document Summarization by Machine Learning Technique for Clustered Documents," 2017.

[38] Z. JIAMING, "Exploiting Textual Structures of Technical Papers for Automatic Multi-document Summarization," 2008.

[39] K. McKeown and D. R. Radev, "Generating summaries of multiple news articles," in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, 1995, pp. 74-82.

[40] D. R. Radev, "A common theory of information fusion from multiple text sources step one: cross-document structure," in Proceedings of the 1st SIGdial workshop on Discourse and dialogue-Volume 10, 2000, pp. 74-83.

[41] O. Bodenreider, "The unified medical language system (UMLS): integrating biomedical terminology," Nucleic acids research, vol. 32, pp. D267-D270, 2004.

[42] N. Elhadad, M.-Y. Kan, J. L. Klavans, and K. R. McKeown, "Customization in a unified framework for summarizing medical literature," Artificial intelligence in medicine, vol. 33, pp. 179-198, 2005.

[43] K. Sarkar, "Using domain knowledge for text summarization in medical domain," International Journal of Recent Trends in Engineering, vol. 1, pp. 200-205, 2009.

[44] K. Hong, "Content selection in multi-document summarization," 2015.

[45] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," Text Summarization Branches Out, 2004.

[46] C.-Y. Lin, "Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough?," in NTCIR, 2004.