Princess Sumaya University for Technology
Since 1991 تأسست
PSUT
جامعة الأميرة سمية للتكنولوجيا

صندوق دعم البحث العلمي
Scientific Research Support Fund

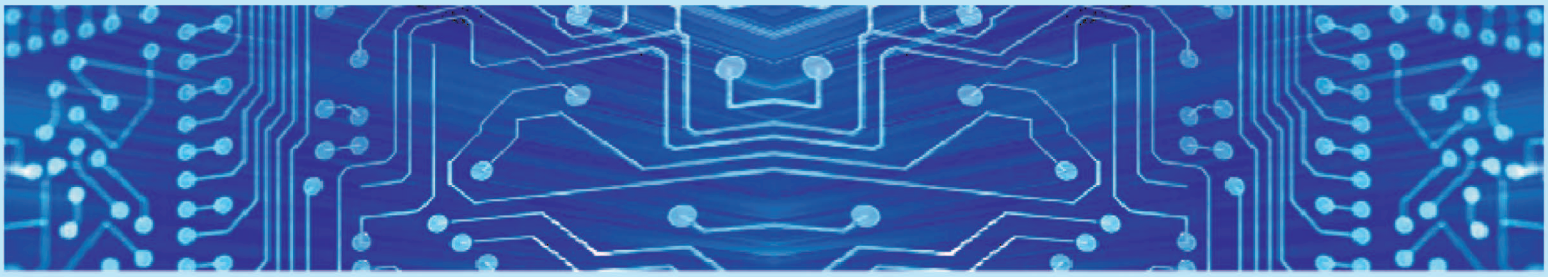
Jordanian Journal of Computers and Information Technology

March 2020 VOLUME 06 NUMBER 01 ISSN 2415 - 1076 (Online) ISSN 2413 - 9351 (Print)

JJCIT

PAGES PAPERS

www.jjcit.org jjcit@psut.edu.jo



An International Peer-Reviewed Scientific Journal
Financed by the Scientific Research Support Fund

Jordanian Journal of Computers and Information Technology (JJCIT)

The Jordanian Journal of Computers and Information Technology (JJCIT) is an international journal that publishes original, high-quality and cutting edge research papers on all aspects and technologies in ICT fields.

JJCIT is hosted by Princess Sumaya University for Technology (PSUT) and supported by the Scientific Research Support Fund in Jordan. Researchers have the right to read, print, distribute, search, download, copy or link to the full text of articles. JJCIT permits reproduction as long as the source is acknowledged.

Aims and Scope

The JJCIT aims to publish the most current developments in the form of original articles as well as review articles in all areas of Telecommunications, Computer Engineering and Information Technology and make them available to researchers worldwide. The JJCIT focuses on topics including, but not limited to: Computer Engineering & Communication Networks, Computer Science & Information Systems and Information Technology and Applications.

Indexing

JJCIT is indexed in:

ELSEVIER Scopus · DOAJ DIRECTORY OF OPEN ACCESS JOURNALS · crossref Text and Data Mining · OCLC WorldCat · BASE · ACADEMIA · Google scholar

Editorial Board Support Team

JJCIT Address

Website: www.jjcit.org
Email: jjcit@psut.edu.jo
Address: Princess Sumaya University for Technology, Khalil Saket Street, Al-Jubaiha
B.O. Box: 1438 Amman 11941 Jordan
Telephone: +962-6-5359949
Fax: +962-6-7295534

EDITORIAL BOARD

INTERNATIONAL ADVISORY BOARD

"Opinions or views expressed in papers published in this journal are those of the author(s) and do not necessarily reflect those of the Editorial Board, the host university or the policy of the Scientific Research Support Fund".

"ما ورد في هذه المجلة يعبر عن آراء الباحثين ولا يعكس بالضرورة آراء هيئة التحرير أو الجامعة أو سياسة صندوق دعم البحث العلمي".

WATER EVAPORATION ALGORITHM WITH PROBABILISTIC NEURAL NETWORK FOR SOLVING CLASSIFICATION PROBLEMS

Mohammed Alweshah[1], Enas Ramadan[1], Mohammad Hashem Ryalat[1], Muder Almi'ani[2] and Abdelaziz I. Hammouri[1]

(Received: 23-Aug.-2019, Revised: 18-Oct.-2019, Accepted: 3-Nov.-2019)

ABSTRACT

Classification is a crucial step in data mining as it facilitates decision-making in many areas of human activity, such as scientific endeavors, marketing campaigns, biomedical research and industrial applications. The probabilistic neural network (PNN) is widely utilized to solve classification and pattern recognition problems and is considered an effective method for solving such problems. In this paper, we propose an improved PNN model that employs the water evaporation algorithm (WEA) in order to solve classification problems more efficiently. The proposed method is able to obtain classification accuracies that are close to each other across all 11 benchmark tested datasets from the UCI machine-learning repository, which demonstrates the validity of this method (with respect to classification accuracy). The results show that the WEA is better than the firefly algorithm (FA) and biogeography-based optimization (BBO) in terms of both classification accuracy and convergence speed.

KEYWORDS:

Classification problems, Data mining, Metaheuristics, Probabilistic neural networks, Water evaporation algorithm.

1. INTRODUCTION

Data mining is the science of extracting valuable information from huge databases in many fields, such as business, academic research and medical activities, by using automatic search processes that employ statistical and computational techniques. It involves the discovery of meaningful patterns and automatic analysis and quantities' exploration of large datasets in order to identify hidden relationships in data [2]. Data mining is used in prediction, in which some of the variables are used to predict other variables (classification) or in description, in which patterns are identified that can be understood easily by the user (clustering) [3].

Classification is a supervised learning task. The classification process separates data into independent classes, the aim of which is to obtain an accurate prediction of the objective class [4]. Data classification helps in producing a required output that could be used in the future [5]. It is very important, as it facilitates decision-making in many domains, such as science, marketing, biomedicine and business [6, 7]. In the field of data mining, many of the techniques that are used for classification problems depend on artificial intelligence. These techniques include the support vector machine (SVM) [8], naïve Bayes (NB) [9], the neural network (NN) [10]-[11], radial basis function (RBF) [12], logistic regression (LR) [13] among many others [14]-[16].

The NN is based on the biological nervous system [4]. The NN was first introduced by Rosenblatt in the late 1950s [17]. The use of the NN method is not a goal in itself; it should instead be seen as an effective tool and a guaranteed means of arriving at the correct prediction of the future values of a phenomenon or a set of variables in any area of application [18]. There are many types of artificial neural network (ANN), including the NN, multilayer perceptron (MLP), feed-forward neural network (FFNN), extreme learning machine (ELM) and the probabilistic neural network (PNN) [19]-[20] .

1. M. Alweshah, E. Ramadan, M. H. Ryalat and A. I. Hammouri are with Department of Computer Science, Prince Abdullah Bin Ghazi Faculty of Communication and Information Technology,Al-Balqa Applied University, Salt, Jordan. Emails: weshah@bau.edu.jo, enas.ramadan35@gmail.com, ryalat@bau.edu.jo and aziz@bau.edu.jo

2. M. Almi'ani is with Computer Information Systems Department, Al-Hussein Bin Talal University, Ma'an, Jordan. Email: malmiani@my.bridgeport.edu

The PNN is a powerful data mining tool and an algorithm that can be used for a vast number of complex relationships (inputs/outputs). The PNN is a spatial form of NN into which a Bayesian statistical decision rule is incorporated. There are four layers in the PNN: (i) an input layer, (ii) a pattern layer, (iii) a summation layer and (iv) an output layer.

The reason for combining metaheuristic algorithms with NNs to create classification tools such as the PNN is to enhance efficiency and effectiveness and enable the solving of difficult problems more quickly and accurately [21]. There are two main kinds of metaheuristics: single-based and population-based. Single-based metaheuristics include local search (LS), simulated annealing (SA) [22] and tabu search (TS) [23]. Population-based metaheuristics include differential evolution (DE) [24], the particle swarm optimization (PSO) algorithm [25], artificial bee colony (ABC) algorithm [26], genetic algorithm (GA) [27], firefly algorithm (FA) [28], NSGA-II [29]-[30] and many others [31].

However, based on a review of the literature, there appears to be a lack of research related to hybridization approaches, especially with respect to whether they could increase the ability to effectively explore and exploit the search space during the search process in tuning the weights of the parameters until the (near) optimal NN weights are obtained [32]. Generally, the weights are initialized to random probability values and then during the search process, the NN weights are updated and will eventually converge to a local optimum solution. Therefore, metaheuristics has been employed to obtain an optimized NN weight that can be fed to the classifier to obtain better classification accuracy [21].

In addition, researchers have introduced population-based approaches to optimized NN weight problems. The main idea behind the population-based is that the algorithms iteratively improve a number of solutions [21]. However, these approaches have some limitations, such as that they are more concerned with exploration rather than exploitation and have a low convergence speed. Motivated by a new metaheuristic technique, water evaporation algorithm (WEA), which is flexible in nature as a population-based technique, has been widely used to solve optimization problems[1]. The WEA has good exploration and exploitation mechanisms, facilitating the finding of near optimal solutions [21].

In this paper, WEA is used to optimize the weights of the PNN in order to improve the performance of the classification system and enable the system to produce the best possible results based on the weights obtained from the PNN. In addition, the WEA is used to achieve a good balance between exploration and exploitation during the search process and thereby improve convergence speed. The proposed method is tested on 11 benchmark classification problems from the UCI machine-learning repository in order to assess its performance.

The remainder of this paper is organized as follows. First, some background on PNN algorithm and WEA is given in section 2. Next, the details of the proposed method are provided in section 3. Then, the experiments and results are presented in section 4. Finally, the conclusion, together with some suggestions for possible directions for future enhancements, are presented in section 5.

2. BACKGROUND

2.1 Probabilistic Neural Network (PNN)

The PNN, which was introduced by Specht [19], is considered one of the most efficient and effective classification techniques. The PPN relies on an algorithm called kernel discriminant analysis [19], [33]-[35]. The training of a PNN does not require the use of heuristic techniques to search for a local minimum [36]. The PNN is arranged in the form of a four-layer feed-forward network structure that consists of an input layer, pattern layer, summation layer and output layer, as shown in Figure 1.

The PNN has the following advantages over other methods: (i) the training process is fast; (ii) it is more accurate than the MLP-PNN; (iii) it is relatively insensitive to outliers; and (iv) it can generate accurate predicted target probability scores. However, it has two disadvantages: (i) slow execution and (ii) a large memory requirement [19]. The four layers of the PNN network are described below:

- The input layer, where each neuron has a predictive variable and feeds values for each of the neurons in the pattern layer.
- The pattern layer, which has one unit for each training sample that formulates a product of the

input vector x with the weight vector w_i, denoted as $z_i = x.w^t_i$ and then performs the following nonlinear process [37]:

$$\exp\left[\left(\frac{(-\,\mathrm{w_i} - \mathrm{x}).(\mathrm{w_i} - x)^{\mathrm{T}}}{(\,2\alpha^2)}\right)\right] \tag{1}$$

- The summation layer, which aggregates the contribution for each class of inputs and generates a network output as a vector of probabilities [37]:

$$\sum\nolimits_i \left[\left(\frac{(-\,\mathrm{w_i} - \mathrm{x}).(\mathrm{w_i} - x)^{\mathrm{T}}}{(\,2\alpha^2)}\right)\right] \tag{2}$$

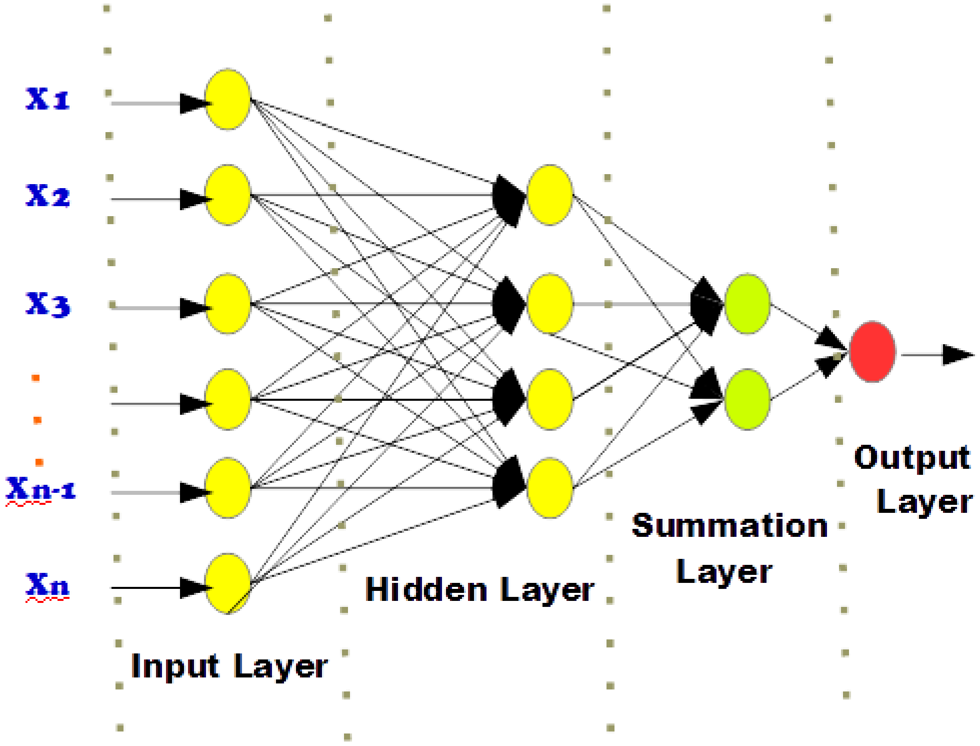


Figure 1. Structure of probabilistic neural network.

- The output layer, which produces binary classes corresponding to the decision classes Ω_r and Ω_s, $r \neq s$, r , $s = 1, 2 ,\ldots\ \ldots,q$ based on a classification criterion:

$$\sum_i \left[\left(\frac{(-\,\mathrm{w_i} - \mathrm{x}).(\mathrm{w_i} - x)^{\mathrm{T}}}{(\,2\alpha^2)}\right)\right] > \sum_j \left[\left(\frac{(-\,\mathrm{w_j} - \mathrm{x}).(\mathrm{w_j} - x)^{\mathrm{T}}}{(\,2\alpha^2)}\right)\right] \tag{3}$$

These nodes have only one weight, C, the prior membership probabilities and the number of training samples in each class, C, given by the cost parameter:

$$C = -\frac{\mathrm{h_s l_s}}{\mathrm{h_r l_r}}.\frac{n_r}{n_s} \tag{4}$$

where, h_s is the preceding prospect, in which the new sample goes to group n and c_n is the misclassification cost [19].

2.2 Water Evaporation Algorithm (WEA)

The WEA is inspired by the natural environment. The basic idea of the WEA is based on simulating the evaporation of a small amount of water molecules located on a solid surface with areas of varied wettability [38]. Solid surfaces are classified according to their behavior toward water into two categories:

1. Hydrophobic (water hating): Such surfaces have low wettability (see Figure 2(b)).
2. Hydrophilic (water loving): Such surfaces have high wettability (see Figure 2(c)).

When water molecules fall on a surface that loves water, such as cotton, water molecules expand on the surface and the evaporation rate is low. However, when water molecules fall on a surface that hates water, such as plastic, water molecules accumulate on the surface in a ball shape and the evaporation rate is high [38].

In the context of this research, a surface with varied wettability can be considered as the search space and water molecules are the solutions within the search space. When surface wettability changes from hydrophilic to hydrophobic, the form of the water molecules changes from a monolayer to a sessile droplet. This is reflected in the layout of the algorithm [38]. The change in the evaporation rate of the water molecules is reflected in the algorithm, where it updates the solutions and this is well matched with the ability of the local and global search algorithm [1].

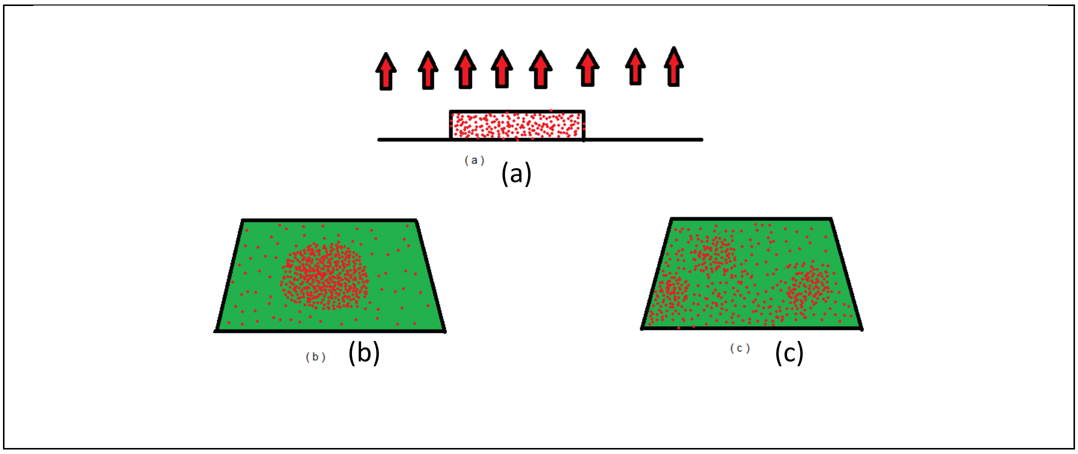


Figure 2. (a) View of initial system; (b) Snapshot of water on a substrate with low wettability; (c) Snapshot of water on a substrate with high wettability [1].

- Locating the initial parameters of the WEA

The number of water molecules (nWM), t_{max}, MEP_{min}, MEP_{max}, DEP_{min} and DEP_{max} and within the search space, the initial locations of all water molecules are created randomly, where t_{max} is maximum number of algorithm iterations, DEP_{min} and DEP_{max} are the minimum and maximum values of the droplet evaporation probability, respectively and MEP_{min} and MEP_{max} are the minimum and maximum values of the monolayer evaporation probability, respectively.

- Generating the water evaporation matrix

There are two approaches that are used to resolve the classification problem: exploration and exploitation.

- **The exploration process**

In the exploration process, if the number of iterations is less than or equal to tmax/2, then a corresponding substrate energy vector is created in order to generate the monolayer evaporation probability matrix by the following equation [38]:

$$E_{sub}(i)^t = \frac{(\mathrm{E_{max}} - \mathrm{E_{min}})*(\mathrm{Fit}_i^t - \mathrm{Min(Fit)})}{(Max(Fit)-Min(Fit))} + E_{min} \tag{5}$$

The water molecules are evaporated globally in accordance with the minimum evaporation probability. The MEP $^{(t)}$ matrix is created by the following equation [38]:

$$MEP_{(t)} = \begin{cases} 1, & if\ rand_{ij} < \exp(E_{sub}(i)^t) \\ 0, & if\ rand_{ij} \geq \exp(E_{sub}(i)^t) \end{cases} \tag{6}$$

- Generating a random permutation-based step-size matrix

The creation of the S matrix is based on a random permutation by the following equation:

$$S = \text{rand} \cdot (\text{WM}(t)[\text{permute1 } (i)(j)] - \text{WM } (t)[\text{permute2 } (i)(j)]) \tag{7}$$

- Generating evaporated water molecules and updating the matrix of water molecules

A set of evaporated water molecules WM $^{(t+1)}$ is created by adding the output of the step-size matrix and the evaporation probability matrix to the current set of molecules by the following equation:

$$\text{WM}^{(t+1)} = \text{WM}^{(t)} + S\times \text{MEP} \tag{8}$$

- **The exploitation process**

In the exploitation process, if the number of iterations is greater than tmax/2, then a contact angle vector is generated in order to produce the droplet evaporation probability (DEP) matrix by the following equation:

$$\theta(i)^t = \frac{(\theta_{max}-\theta_{min})*(Fit_i^t - Min(Fit))}{(Max(Fit)-Min(Fit))} + \theta_{min} \tag{9}$$

Evaporation is based on the droplet evaporation probability. The DEP$^{(t)}$ matrix is created by the following equation:

$$DEP_{ij}^t = \begin{cases} 1, \ if\ rand_{ij} < J(\theta_i^t) \\ 0, ifrand_{ij} \geq J(\theta_i^t) \end{cases} \tag{10}$$

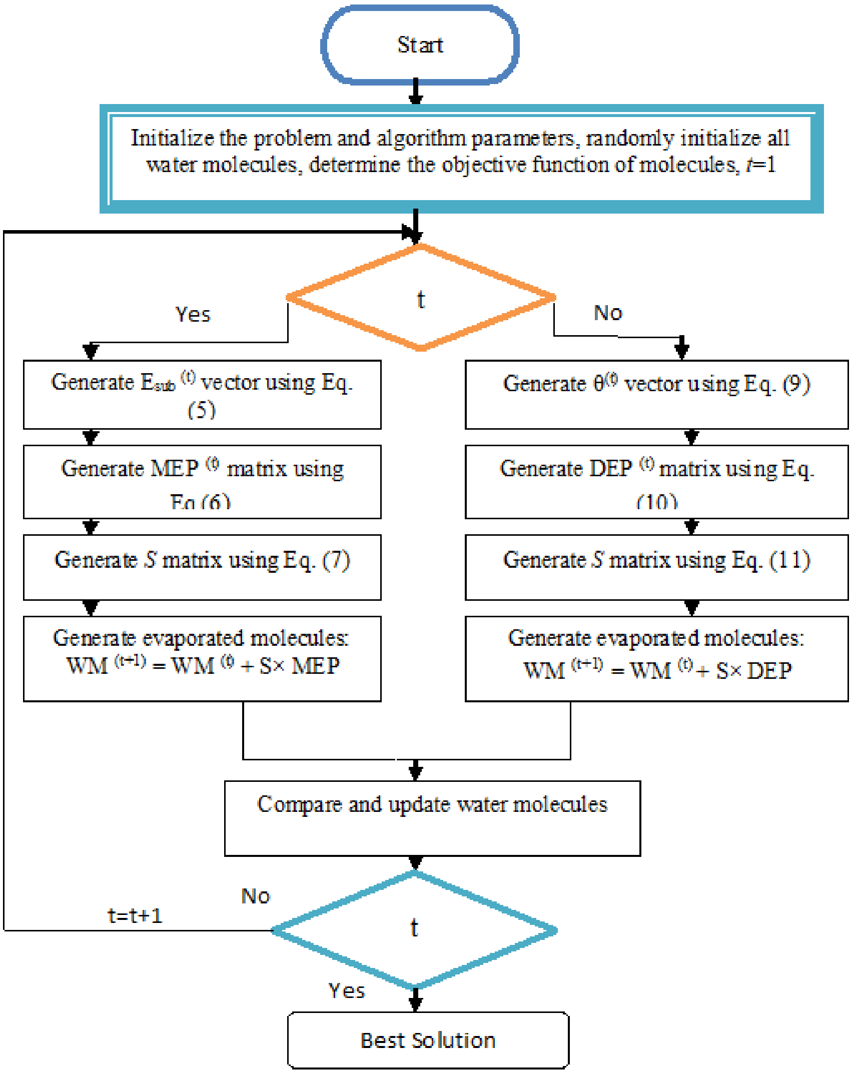

Start
Initialize the problem and algorithm parameters, randomly initialize all water molecules, determine the objective function of molecules, t=1
t
Yes
No
Generate E$_{sub}$ $^{(t)}$ vector using Eq. (5)
Generate MEP $^{(t)}$ matrix using Eq.(6)
Generate S matrix using Eq. (7)
Generate evaporated molecules: WM $^{(t+1)}$ = WM $^{(t)}$ + S× MEP
Generate θ$^{(t)}$ vector using Eq. (9)
Generate DEP $^{(t)}$ matrix using Eq. (10)
Generate S matrix using Eq. (11)
Generate evaporated molecules: WM $^{(t+1)}$ = WM $^{(t)}$+ S× DEP
Compare and update water molecules
No
t=t+1
t
Yes
Best Solution

Figure 3. Water Evaporation Algorithm (WEA).

- Generating a random permutation-based step-size matrix

The creation of the S matrix is based on a random permutation by the following equation:

$$S= \text{rand} \cdot (WM^{(t)} [\text{permute1 } (i)\ (j)] - WM\ (t) [\text{permute2 } (i)\ (j)]) \tag{11}$$

- Generating evaporated water molecules and updating the matrix of water molecules

A set of evaporated water molecules WM $^{(t + 1)}$ is created by adding the output of the step size matrix and the evaporation probability matrix to the current set of molecules by the following equation [38]:

$$WM^{(t+1)} = WM^{(t)} + S \times DEP \tag{12}$$

- Checking whether the termination criterion is met

If the current iteration value is greater than the upper limit of the iterations, then the algorithm terminates. Otherwise, it moves to Step 2 [38].

3. Proposed Method: WEA with PNN

In this research, for the first time the PNN is hybridized with the WEA in an attempt to find a way to solve classification problems more efficiently. Hereinafter, the name WEA-PNN will be used to denote the hybridization of the WEA and the PNN.

Figure 4 shows how the initial weights are generated randomly by the PNN classifier. As determined by the PNN classifier, the values of the input data are multiplied by the corresponding weights w(ij).

As seen from Figure 4, the procedure starts from initial weights that are randomly generated by the original PNN classification model. The values from the input data are then multiplied by the appropriate weights $w_{(ij)}$, as determined by the PNN algorithm shown in Figure 5 and transmitted to the pattern layer as in Equation 3. Latter are converted through a transfer function [19] into summation and output layers as in Equation 4. The output layer typically contains only one class, since only one output is usually requested. During the training phase, the goal is to determine the most accurate weights to be assigned to the connector line. Furthermore, during the training, the output is computed repeatedly and the result is compared to the preferred output generated by the training/testing datasets.

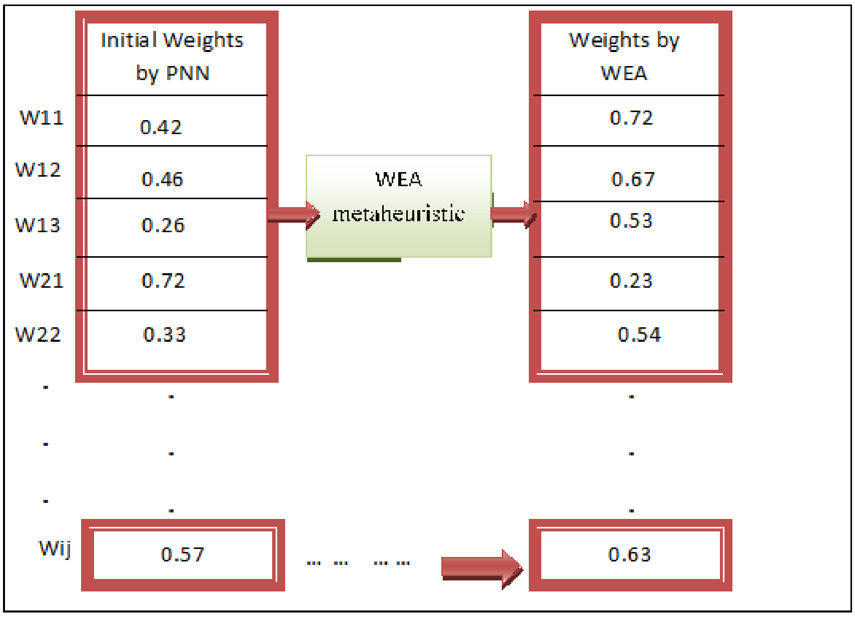


Figure 4. Representation of initial weights.

The second part of the proposed approach is the (FA) which has been used as an improvement algorithm. The firefly algorithm is one of the efficient methods of solving complex problems. Figure 5 illustrates the steps involved in applying the WEA with the PNN. It consists of two parts: the first part is the PNN which uses the training data and also classifies the tested data. After that, the WEA is used to adjust the PNN weights. Then, the accuracy of the classified data procedure is repeated until the termination criterion is met.

The classification quality of the proposed technique is measured by calculating the accuracy value as in Equation 13, where accuracy is calculated based on the number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) results.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{13}$$

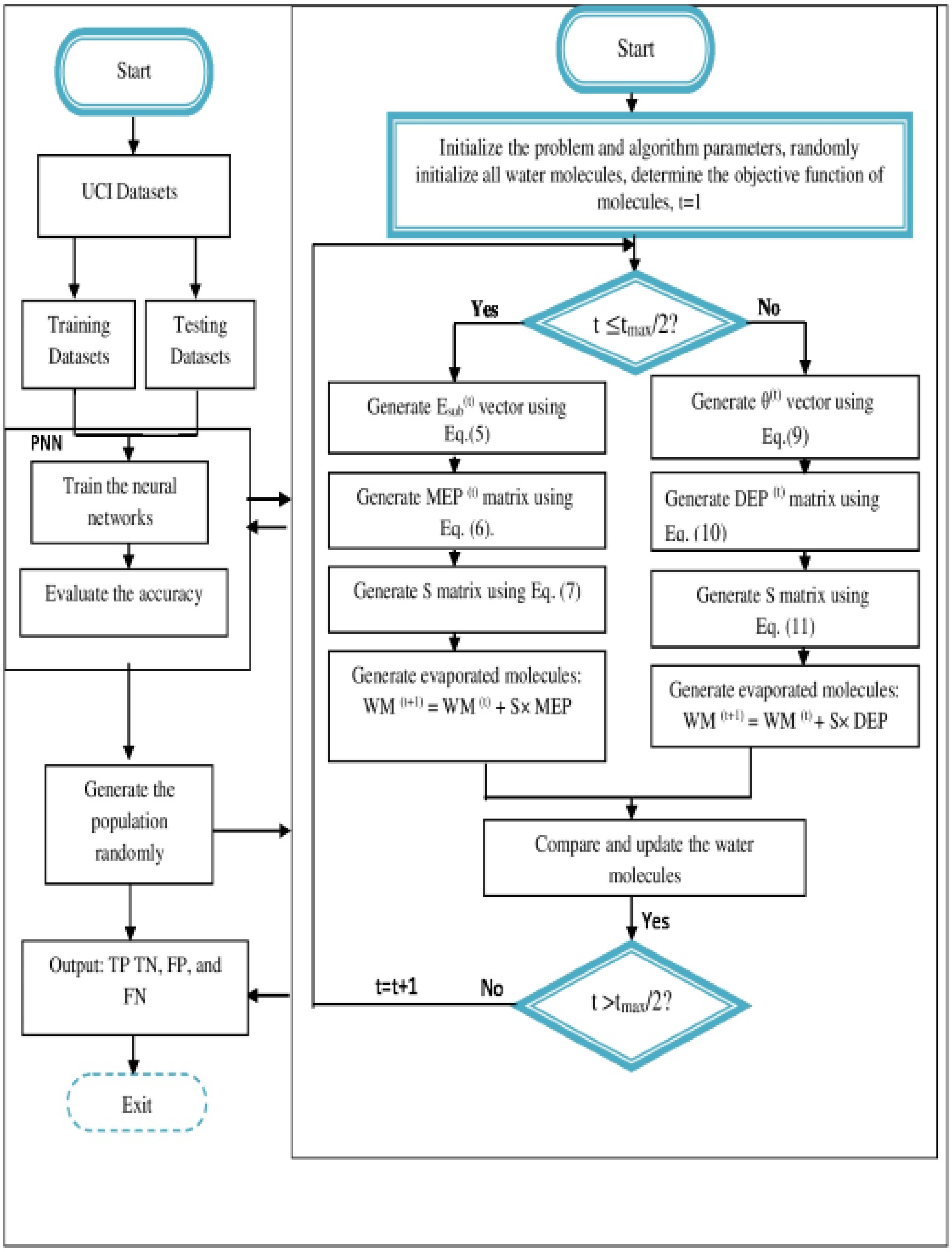

Start
UCI Datasets
Training Datasets
Testing Datasets
PNN
Train the neural networks
Evaluate the accuracy
Generate the population randomly
Output: TP TN, FP, and FN
Exit
Start
Initialize the problem and algorithm parameters, randomly initialize all water molecules, determine the objective function of molecules, t=1
Yes
No
t ≤t_max/2?
Generate E_sub^(t) vector using Eq.(5)
Generate MEP ^(t) matrix using Eq. (6).
Generate S matrix using Eq. (7)
Generate evaporated molecules: WM ^(t+1) = WM ^(t) + S× MEP
Generate θ^(t) vector using Eq.(9)
Generate DEP ^(t) matrix using Eq. (10)
Generate S matrix using Eq. (11)
Generate evaporated molecules: WM ^(t+1) = WM ^(t) + S× DEP
Compare and update the water molecules
Yes
t=t+1
No
t >t_max/2?

Figure 5. Flowchart of WEA-PNN technique.

If both the predicted label and the object's actual label are positive, the class is classified as TP. If both the predicted label and the object's actual label are negative, it is classified as TN. On the other hand, the class is classified as FP when the predicted class is positive but the actual label is negative. It is classified as FN when the predicted class is negative but the actual label is positive. These four counts are presented in Table 1 for the binary classification [39]:

Table 1. Cross-matrix classification.

		Predicted class	
Actual class		**Positive**	**Negative**
	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

Three other performance measurements were also calculated to assess the performance of the proposed WEA-PNN: sensitivity (Equation 14), specificity (Equation 15) and G-mean (Equation 16). The error rate was also obtained (Equation 17).

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \tag{14}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \tag{15}$$

$$\text{G} - \text{mean} = \sqrt{(\text{Sensitivity} \times \ \text{Specificity})} \tag{16}$$

$$\text{Error Rate} = 1 - \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{17}$$

4. Experiments and Results

Experiments were conducted to test the proposed technique by using Matlab R2010a on an Intel ® Xeon ®CPU ES-1630 v3 @3.70 GH$_z$ computer with 16 GB RAM and a Windows 10 operating system. The input parameters that were used for all the experiments and datasets are shown in Table 2.

Table 2. Input parameter setting.

TMax	100
ThetaMax	50
ThetaMin	20
MepMin	0.03
MepMax	0.6
Number of iterations	200
Population size (# of water molecules)	50

After 30 autonomous runs for each of the 11 datasets that can be freely downloaded from http://csc.lsu.edu/~huypham/HBA_CBA/datasets.html , the solutions were provided in terms of best accuracy. When the accuracy is 100% and the error is zero, we get the best results. In this situation, the number of FPs and FNs would be 0 and the number of TPs and TNs would be the total number of observed positive classes and the total number of observed negative classes.

Table 3 presents the results of the WEA-PNN when applied to the 11 selected datasets together with the results for the basic PNN and the results that were reported in the literature for the FA [36] and biogeography-based optimization (BBO) [15] with PNN in terms of accuracy, sensitivity, specificity, error rate (%) and ratio G-mean.

Table 3. Classification accuracy, sensitivity, specificity, error rate and ratio G-mean for PNN and for FA, BBO and WEA with PNN.

Dataset	Model	TP	FP	TN	FN	Accuracy	Sensitivity	Specificity	Ratio G-mean
PID	PNN	35	28	90	39	65.104	47.30	76.27	60.06
	FA-PNN	33	30	113	16	76.040	*67.35*	79.02	72.95
	BBO –PNN	38	25	99	30	71.350	55.88	79.84	66.79
	WEA-PNN	39	24	122	7	**83.854**	**84.78**	**85.71**	**85.24**
HSS	PNN	44	12	6	15	64.93	74.58	33.33	49.86
	FA-PNN	54	2	10	11	*83.12*	83.08	***83.33***	**83.20**
	BBO –PNN	52	4	11	10	81.82	*83.87*	73.33	78.42
	WEA-PNN	53	3	12	9	**84.42**	**85.48**	80.00	82.69
AP	PNN	23	1	1	2	88.88	92.00	50.00	67.82
	FA-PNN	24	0	1	2	**92.59**	**92.31**	**100.00**	**96.08**
	BBO –PNN	52	4	11	10	81.82	*83.87*	73.33	78.42
	WEA-PNN	24	0	1	2	**92.59**	**92.31**	**100.00**	**96.08**
BC	PNN	14	9	36	13	69.44	51.9	80.00	64.44
	FA-PNN	31	1	24	12	80.88	72.09	**96.00**	**83.19**
	BBO–PNN	13	10	44	5	79.17	72.22	81.48	76.71
	WEA-PNN	17	6	44	5	**84.72**	**77.27**	88.00	82.46
LD	PNN	18	15	34	19	60.46	48.60	69.40	58.08
	FA-PNN	31	1	24	12	79.07	72.09	96.00	83.19
	BBO–PNN	32	0	23	13	72.09	71.11	**100.0**	84.30
	WEA-PNN	24	9	49	4	**84.88**	**85.71**	84.48	**85.09**
Heart	PNN	27	5	23	13	73.53	67.50	82.10	74.44
	FA-PNN	31	1	24	12	80.88	72.09	96.00	83.19
	BBO–PNN	32	0	23	13	80.90	71.11	**100.00**	84.33
	WEA-PNN	32	0	25	11	**83.82**	**74.42**	**100.00**	**86.27**
GCD	PNN	133	46	39	32	68.80	80.60	45.90	60.82
	FA-PNN	166	13	30	41	78.40	80.19	69.77	74.79
	BBO–PNN	139	40	44	27	73.20	83.73	52.38	66.23
	WEA-PNN	165	14	42	29	**82.80**	**85.05**	**75.00**	**79.87**
Parkinsons	PNN	39	0	4	6	87.75	86.67	100.00	93.09
	FA-PNN	38	1	6	4	89.80	90.48	85.71	88.06
	BBO–PNN	39	0	7	3	**93.88**	**92.86**	**100.00**	**96.36**
	WEA-PNN	93	0	7	3	**93.88**	**92.86**	**100.00**	**96.36**
SPECTF	PNN	49	4	5	9	80.59	84.48	55.56	68.51
	FA-PNN	52	1	10	4	**92.54**	92.86	**90.91**	**91.88**
	BBO–PNN	49	9	5	5	86.57	90.74	35.71	56.92
	WEA-PNN	49	4	12	2	91.04	**96.08**	75.00	84.88
ACA	PNN	60	14	84	15	83.24	80.00	85.70	82.80
	FA-PNN	65	9	94	5	91.91	92.86	91.26	92.06
	BBO –PNN	65	9	88	11	88.53	85.53	90.72	88.09
	WEA-PNN	71	3	94	5	**95.38**	**93.42**	**96.91**	**95.15**
Fourclass	PNN	59	19	127	11	86.11	84.29	86.99	85.63
	FA-PNN	78	0	138	0	**100.00**	**100.00**	**100.00**	**100.00**
	BBO–PNN	78	0	138	0	**100.00**	**100.00**	**100.00**	**100.00**
	WEA-PNN	78	0	138	0	**100.00**	**100.00**	**100.00**	**100.00**

It can be seen from the table that WEA-PNN outperformed the other methods in terms of accuracy and had superiority in 10 out of the 11 datasets. The best results are presented in bold. The standard deviations and accuracy means of the proposed technique are presented in Table 4. For example, in the

PIMA Indian diabetes (PID) dataset, the original PNN has achieved 65.1% accuracy rate, while the proposed WEA-PNN obtained 83.85% accuracy rate.

In other words, it has good exploitation capability and can find better solutions as many candidates are gathered near optimal solution. The proposed WEA shows better performance (with respect to accuracy, sensitivity, specificity and error rate) than the original PNN algorithm on almost all datasets.

The P-value is the estimated probability of rejecting the null hypothesis (H0, no difference between two groups) when that hypothesis is true. The alternative hypothesis (H1) is the opposite of the null hypothesis. The significance level (α) in t-test is used to refer to a pre-chosen probability. If the calculated P-value is less than the chosen significance level, this provides reasonable evidence to support the alternative hypothesis and reject the null hypothesis. The choice of α or significance level to reject the null hypothesis is arbitrary. Most of researchers refer to statistical significance when P-value < 0.05; more details about t-test can be found in the book “Introduction to Probability and Statistics” [40] .

The performance of the WEA was further verified by determining whether it was statistically different from the FA. This was done by using a t-test with a significance interval of 95% (α = 0.05) for classification accuracy. Table 4 presents the accuracy statistics for the WEA and the FA. From the table, it can be seen that the performance of the WEA is much better than that of the FA, because all the P-values are less than 0.01.

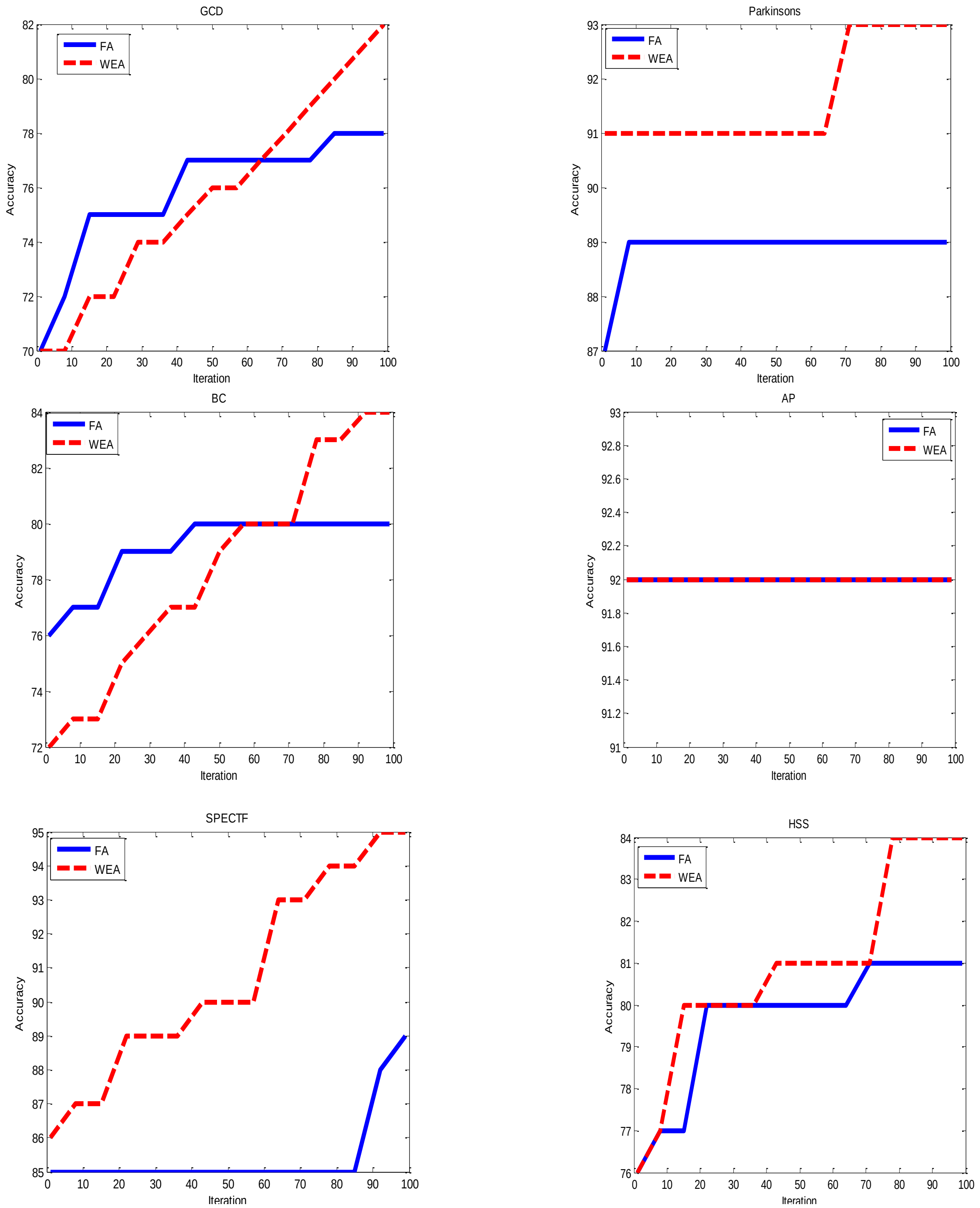
Table 4. The statistics and P-values of the t-test for the accuracy of the WEA and FA.

Dataset		Mean	Std. deviation	Std. error mean	P-value
PID	WEA	83.1028	0.63377	0.11570	0.00
	FA	73.4895	1.28560	0.23472	
HSS	WEA	83.3766	0.62887	0.11482	0.00
	FA	81.8179	1.02322	0.18681	
AP	WEA	92.5926	0.00000	0.00000	0.00
	FA	92.5926	0.00012	0.00002	
BC	WEA	82.9167	1.59613	0.29141	0.00
	FA	77.3935	1.74347	0.31831	
LD	WEA	83.3333	2.23006	0.40720	0.00
	FA	75.5810	1.49604	0.27310	
Heart	WEA	82.2549	1.80867	0.33022	0.00
	FA	78.6819	2.23781	0.40857	
GCD	WEA	82.8000	0.00000	0.00000	0.00
	FA	75.1600	1.58040	0.28854	
Parkinson's	WEA	92.5170	1.72282	0.31450	0.00
	FA	89.7950	0.00000	0.00000	
SPECTF	WEA	88.2668	1.37125	0.25035	0.00
	FA	88.8057	1.82787	0.33372	
ACA	WEA	94.5087	0.75519	0.13788	0.00
	FA	89.8840	1.05983	0.19350	
Fourclass	WEA	100.000	0.00000	0.00000	0.00
	FA	100.000	0.00000	0.00000	

Figure 6 illustrates the simulation outcomes of the convergence characteristics of the FA-PNN and the WEA-PNN when they were applied to the 11 datasets. Each algorithm was run for 200 iterations. The experimental results indicate that the WEA has a faster convergence than the FA. Moreover, for the AP dataset, the WEA produced a comparable convergence trend to that of the FA. Interestingly, the WEA accomplished 100% accuracy in all iterations when applied to the Fourclass dataset.

Figure 7 shows box plots that illustrate the distribution of the resolution quality obtained by the WEA and the FA for the 11 datasets.

In brief, the simulation results confirm and indicate that hybrid method WEA is one of the suitable methods for classification problems, since it shows a good performance, where it is not ranked last for all the tested datasets (with respect to the classification accuracy).



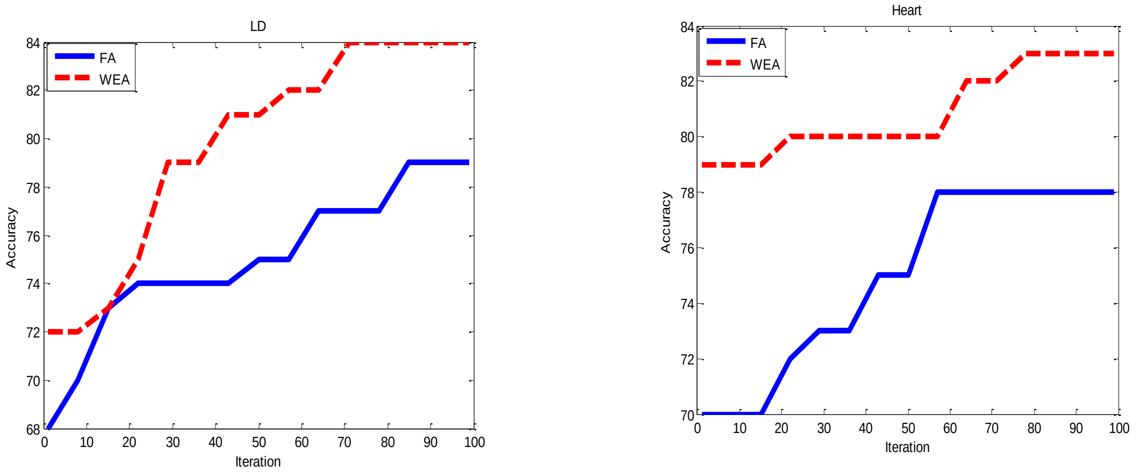


Figure 6. Convergence characteristics of WEA and FA.

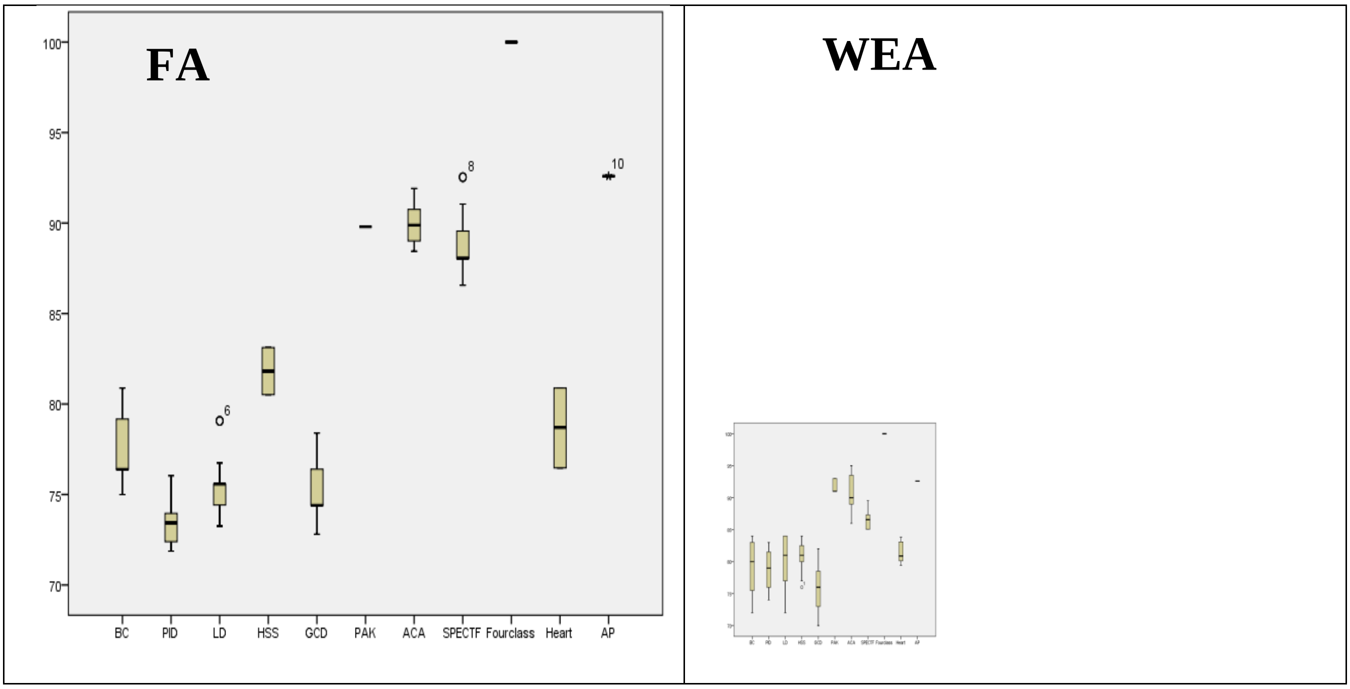


Figure 7. Box plots for FA and WEA.

5. Conclusion

The overall purpose of this work was to investigate the performance of the WEA in solving classification problems. This paper presented the outcomes of applying the proposed approach (WEA-PNN) to 11 benchmark datasets from the UCI machine-learning repository. The performance of the WEA-PNN was assessed in terms of classification accuracy and convergence speed. The outcomes were also analyzed by using the t-test to assess the accuracy obtained for all the datasets. Then, the outcomes of WEA-PNN were compared with those of other methods in the literature. The results indicated that the suggested technique was able to obtain the best convergence speed and higher accuracy (i.e., having superiority in 10 out of 11 datasets) than the compared methods. This shows that the WEA-PNN is capable of generating better outcomes than those produced by other techniques in the literature.

A stable and fast convergence can lead to better solutions. It can be done for example by using a "randomized" greedy heuristic to obtain different initial solutions (in the initial population) rather than a random initialization that may lose its diversity which later will generate a premature convergence and stagnation of the population. This is subject to the further enhanced work.

REFERENCES

[1] A. Kaveh and T. Bakhshpoori, "Water Evaporation Optimization: A Novel Physically Inspired Optimization Algorithm," Computers & Structures, vol. 167, pp. 69-85, 2016.

[2] D. Li and Y. Du, Artificial Intelligence with Uncertainty, CRC press, 2017.

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

[4] H. Faris, I. Aljarah and S. Mirjalili, "Training Feedforward Neural Networks Using Multi-verse Optimizer for Binary Classification Problems," Applied Intelligence, vol. 45, pp. 322-332, 2016.

[5] O. Adwan, H. Faris, K. Jaradat, O. Harfoushi and N. Ghatasheh, "Predicting Customer Churn in Telecom Industry Using Multilayer Preceptron Neural Networks: Modeling and Analysis," Life Science Journal, vol. 11, pp. 75-81, 2014.

[6] R. L. Schalock, S. A. Borthwick-Duffy, V. J. Bradley, W. H. Buntinx, D. L. Coulter, E. M. Craig, S. C. Gomez, Y. Lachapelle, R. Luckasson and A. Reeve, Intellectual Disability: Definition, Classification and Systems of Supports, ERIC, 2010.

[7] T. R. Kiran and S. Rajput, "An Effectiveness Model for An Indirect Evaporative Cooling (IEC) System: Comparison of Artificial Neural Networks (ANN), Adaptive Neuro-Fuzzy Inference System (ANFIS) and Fuzzy Inference System (FIS) Approach," Applied Soft Computing, vol. 11, pp. 3525-3533, 2011.

[8] Z. Yang, Z. I. Rauen and C. Liu, "Automatic Tuning on Many-Core Platform for Energy Efficiency *via* Support Vector Machine Enhanced Differential Evolution," Scalable Computing: Practice and Experience, vol. 18, pp. 117-132, 2017.

[9] N. Friedman, D. Geiger and M. Goldszmidt, "Bayesian Network Classifiers," Machine learning, vol. 29, pp. 131-163, 1997.

[10] A. Alshareef, S. Ahmida, A. A. Bakar, A. R. Hamdan and M. Alweshah, "Mining Survey Data on University Students to Determine Trends in the Selection of Majors," Proc. of the Science and Information Conference (SAI), pp. 586-590, 2015.

[11] A. Alshareef, A. Alkilany, M. Alweshah and A. A. Bakar, "Toward a Student Information System for Sebha University, Libya," Proc. of the 5th International Conference on Innovative Computing Technology (INTECH), pp. 34-39, 2015.

[12] W.-H. Au and K. C. Chan, "Classification with Degree of Membership: A Fuzzy Approach," Proceedings of IEEE International Conference on Data Mining (ICDM 2001), pp. 35-42, 2001.

[13] E. M. Azoff, Neural Network Time Series Forecasting of Financial Markets, John Wiley & Sons, Inc., 1994.

[14] M. Alweshah, A. I. Hammouri, H. Rashaideh, M. Ababneh and H. Tayyeb, "Solving Time Series Classification Problems Using Combined of Support Vector Machine and Neural Network," International Journal of Data Analysis Techniques and Strategies, vol. 9, 2017.

[15] M. Alweshah, A. I. Hammouri and S. Tedmori, "Biogeography-based Optimization for Data Classification Problems," International Journal of Data Mining, Modelling and Management, vol. 9, pp. 142-162, 2017.

[16] M. Alweshah, M. Abu Qadoura, A. I. Hammouri, M. S. Azmi and S. Alkhalaileh, "Flower Pollination Algorithm for Solving Classification Problems," International Journal of Advances in Soft Computing and Its Applications, In Press, pp. 1-13, 2019.

[17] M. Alweshah, "Firefly Algorithm with Artificial Neural Network for Time Series Problems," Research Journal of Applied Sciences, Engineering and Technology, vol. 7, pp. 3978-3982, 2014.

[18] I. Aljarah, H. Faris and S. Mirjalili, "Optimizing Connection Weights in Neural Networks Using the Whale Optimization Algorithm," Soft Computing, vol. 22, pp. 1-15, 2018.

[19] D. F. Specht, "Probabilistic Neural Networks," Neural networks, vol. 3, pp. 109-118, 1990.

[20] M. Alweshah, "Construction Biogeography-based Optimization Algorithm for Solving Classification Problems," Neural Computing and Applications, pp. 1-10, 2018.

[21] E.-G. Talbi, Metaheuristics: From Design to Implementation, John Wiley & Sons, 2009.

[22] Tareq Aziz AL-Qutami, I. I. Rosdiazli Ibrahim and M. A. Ishak, "Virtual Multiphase Flow Metering

Using Diverse Neural Network Ensemble and Adaptive Simulated Annealing," Expert Systems with Applications, vol. 93, pp. 72-85, 2017.

[23] F. Glover and M. Laguna, "Tabu Search∗," Handbook of Combinatorial Optimization, Ed., Springer, , pp. 3261-3362, 2013.

[24] A. Slowik and M. Bialko, "Training of Artificial Neural Networks using Differential Evolution Algorithm," Proc. of the Conference on Human System Interactions, pp. 60-65, 2008.

[25] J. Kennedy, "Particle Swarm Optimization," Encyclopedia of Machine Learning, Ed., Springer, pp. 760-766, 2011.

[26] X.-S. Yang, Nature-inspired Metaheuristic Algorithms, Luniver Press, 2010.

[27] D. J. Montana and L. Davis, "Training Feedforward Neural Networks Using Genetic Algorithms," IJCAI, pp. 762-767, 1989.

[28] X. S. Yang, "Firefly Algorithm," Engineering Optimization, pp. 221-230, 2010.

[29] R. Ak, Y. Li, V. Vitelli, E. Zio, E. L. Droguett and C. M. C. Jacinto, "NSGA-II-trained Neural Network Approach to the Estimation of Prediction Intervals of Scale Deposition Rate in Oil & Gas Equipment," Expert Systems with Applications, vol. 40, pp. 1205-1212, 2013.

[30] Z. Hongwu, Z. Jinya, L. Yan and Y. Chun, "Multi-objective Optimization of Helico-axial Multiphase Pump Impeller Based on NSGA-II," Proc. of the 2nd International Conference on Intelligent Computation Technology and Automation, pp. 202-205, 2009.

[31] E. Solgi, S. M. M. Husseini, A. Ahmadi and H. Gitinavard, "A Hybrid Hierarchical Soft Computing Approach for the Technology Selection Problem in Brick Industry Considering Environmental Competencies: A Case Study," Journal of Environmental Management, vol. 248, p. 109219, 2019.

[32] D. Whitley, T. Starkweather and C. Bogart, "Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity," Parallel Computing, vol. 14, pp. 347-361, 1990.

[33] K. Z. Mao, K.-C. Tan and W. Ser, "Probabilistic Neural-network Structure Determination for Pattern Classification," IEEE Transactions on Neural Networks, vol. 11, pp. 1009-1016, 2000.

[34] V. Kwigizile, M. F. Selekwa and R. N. Mussa, "Highway Vehicle Classification by Probabilistic Neural Networks," Proc. of the FLAIRS Conference, pp. 664-669, 2004.

[35] W. P. Sweeney Jr, M. T. Musavi and J. N. Guidi, "Classification of Cromosomes Using a Probabilistic Neural Network," Cytometry: The Journal of the International Society for Analytical Cytology, vol. 16, pp. 17-24, 1994.

[36] M. Alweshah and S. Abdullah, "Hybridizing Firefly Algorithms with a Probabilistic Neural Network for Solving Classification Problems," Applied Soft Computing, vol. 35, pp. 513-524, 2015.

[37] P. Wasserman, Advanced Methods in Neural Networks: van Nostrand Reinhold, 1993.

[38] A. Saha, P. Das and A. K. Chakraborty, "Water Evaporation Algorithm: A New Metaheuristic Algorithm Towards the Solution of Optimal Power Flow," Engineering Science and Technology, an International Journal, vol. 20, pp. 1540-1552, 2017.

[39] M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," Information Processing & Management, vol. 45, pp. 427-437, 2009.

[40] W. Mendenhall III, R. J. Beaver and B. M. Beaver, Introduction to Probability and Statistics: Cengage Learning, 2013.

ملخص البحث:

يُعدّ التصنيف خطوة حاسمة في تعدين البيانات؛ لأنّه يسهل اتخاذ القرار في مجالات عدّة من النشاط الإنساني، مثل الجهود العلمية، وحملات التسويق، والبحث الطبي الحيوي والتطبيقات الصناعية. ويستفاد من الشبكة العصبية الاحتمالية على نطاق واسع في حل مشكلات التصنيف وتمييز الأنماط؛ فهي تعدّ طريقة فعالة لحل مثل هذه المشكلات.

في هذه الورقة، نقترح نموذجاً محسَّناً لشبكة عصبية احتمالية تستخدم خوارزمية تبخر الماء (WEA) لحل مشكلات التصنيف على نحوٍ أكثر فاعلية. وتتميز الطريقة المقترحة بقدرتها على تحقيق مستويات متقاربة من الدقة عند تطبيقها على إحدى عشرة مجموعة بيانات مرجعية من مخزون تعلم الآلة، الأمر الذي يؤكد نجاعتها من حيث دقة التصنيف.

وتبين النتائج أن خوارزمية تبخر الماء كانت أفضل من خوارزمية اليَراعة (FA)، ومن الأمثلة القائمة على الجغرافيا الحيوية (BBO) من حيث دقة التصنيف وسرعة التقارب.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 01, March 2020.

TRANSMIT ANTENNA SELECTION SCHEMES FOR DOUBLE SPATIAL MODULATION

Belal A. Asaati[1] and Ammar M. Abu-Hudrouss [2]

(Received: 3-Sep.-2019, Revised: 22-Oct.-2019, Accepted: 6-Nov.-2019)

ABSTRACT

Double spatial modulation (DSM) is a transmission technique which has been recently proposed for multiple-input multiple-output (MIMO) communication systems. DSM has a higher spectral efficiency compared with classical spatial modulation (SM), as it doubles the number of active transmit antennas. In this paper, transmit antenna selection (TAS) is applied to DSM in order to enhance the bit error rate (BER) performance. In particular, we integrate two sub-optimal TAS algorithms to DSM; namely, capacity-optimized antenna selection (COAS) and antenna selection based on amplitude and antenna correlation (A-C-AS). Simulation results of these two algorithms are presented and compared with the optimal Euclidean distance-optimized antenna selection (EDAS) using MATLAB software. Our results show a complexity-performance trade-off. Although there is a negligible loss of BER, our algorithms are much less complex than EDAS.

KEYWORDS

Antenna selection, Double spatial modulation, Transmit diversity, Spectral efficiency, Bit error rate.

1. INTRODUCTION

In MIMO systems, spatial multiplexing is used to serve the need for higher data rates in wireless communications. It utilizes multiple transmitting antennas in order to convey the data simultaneously [1]. One popular example for spatial multiplexing in MIMO is the vertical Bell lab layered space-time (VBLAST) scheme [2]-[3]. VBLAST achieves a high data rate, but suffers from inter-channel interference (ICI) and high receiver complexity.

To overcome the pitfalls of the VBLAST, spatial modulation (SM) is another transmission scheme that overcomes the problem of the ICI and has a lower receiver complexity than the VBLAST [4]-[5]. SM is a member of the index modulation family which attracted an increased attention in the past decade [6]-[8]. Although SM improves the spectral efficiency of MIMO systems, it does not achieve the same data rate of the VBLAST. Therefore, the improvement of spectral efficiency under SM has been achieved through several schemes [6]. Examples of the recent schemes of SM include quadrature spatial modulation (QSM) [9] and double spatial modulation (DSM) [10]. QSM retains the benefits of SM, but with an improved spectral efficiency. The basic idea of QSM is to split the in-phase and quadrature components of the amplitude/ phase modulation (APM) symbol and map them separately to the antenna set [9]. Meanwhile, DSM allows transmitting two modulated symbols at the same time. DSM provides considerably better error performance than QSM [10]. Moreover, the spectral efficiency of the classical SM is a half of the spectral efficiency of the DSM scheme for the same number of transmit antennas and modulation order, M.

Despite the several advantages of SM-MIMO systems, combining transmit diversity with these systems is not straightforward [11]. Antenna selection schemes (TASs) can be used to introduce transmit diversity for SM systems [12]-[15]. In [12], a tree search antenna selection scheme (TSAS) for SM systems is introduced to reduce the high complexity of EDAS scheme. In [13], a low complexity TAS algorithm based also on Euclidian distance is presented. Several sub-optimal TAS schemes are used to enhance the performance of QSM in [14]. The performance is compared with the optimal EDAS. The suboptimal schemes have much lower complexity compared with EDAS with a reasonable deterioration in bit error rate (BER) performance [15]. Up to the author knowledge, the performance of antenna selection schemes has not been studied with DSM.

B. A. Asaati and A. M. Abu-Hudrouss are with Islamic University of Gaza, Gaza, Palestine. Emails: [1]engbelal2050t@gmail.com and [2]ahdrouss@iugaza.edu.ps

The contribution of this paper is to introduce transmit diversity for the DSM. Moreover, it studies the impacts of antenna selection algorithms on DSM. The performance of the applied TAS algorithms is analyzed in terms of both computational complexity and BER probability.

The paper is organized as follows: Section 2 describes the DSM transmission technique. Different TAS schemes are discussed in Section 3. Monte-Carlo simulation results and comparisons are provided in Section 4 and the paper's conclusions are given in Section 5.

2. DSM Transceiver

In the DSM transceiver shown in Figure 1 and Figure 2, the input binary bits $m = \log_2(N_t^2 M^2)$ are split into two equal parts by a primary splitter, each containing $\log_2(N_t M)$ bits, where N_t and M are the total number of transmit antennas and constellation size, respectively. Consequently, each part selects its own information symbol and the position of the active transmit antenna. Therefore, the $\log_2(N_t M)$ bits are split into two sets of bits by a secondary splitter. The first set of bits, $\log_2(N_t)$, determines the location of an active transmit antenna, whilst the second set of bits, $\log_2(M)$, determines the corresponding transmit symbol from *M*-ary signal constellation.

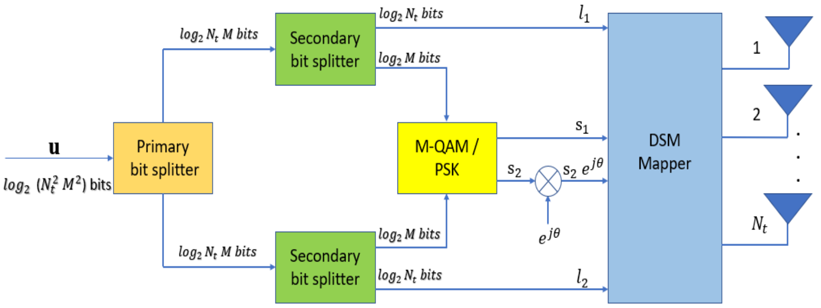


Figure 1. Block diagram of DSM transmitter.

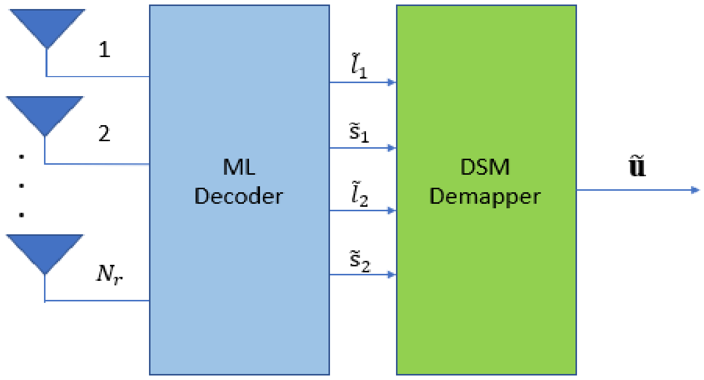


Figure 2. Block diagram of DSM receiver.

A DSM transmission vector is constructed by the superposition of two independent SM transmission vectors [10]. One of the information symbols s_1 is sent through its corresponding activated transmit antenna l_1, while the second information symbol s_2 is sent through the second active antenna l_2 with a rotation angle θ. The rotation angle θ is optimized for *M*-ary signal constellation to distinguish the two information symbols from each other and to decrease the BER [10].

Therefore, under DSM, the transmitted vector **s** of size of $N_t \times 1$ is given by [10]. Generally, the spectral efficiency of DSM is given as follows [10]:

$$\mathbf{s} = \left[0 \cdots 0 \ \underbrace{s_1}_{l_1} \ 0 \cdots 0 \ \underbrace{s_2 \, e^{j\theta}}_{l_2} \ 0 \cdots 0 \right]^T \tag{1}$$

$$m = \log_2(N_t^2) + \log_2(M^2), \tag{2}$$

The received vector $\mathbf{y}$, of size $N_r \times 1$, can be expressed as:

$$\mathbf{y} = \mathbf{Hs} + \mathbf{n} = \mathbf{h}_{l_1} s_1 + \mathbf{h}_{l_2} s_2 e^{j\theta} + \mathbf{n} \tag{3}$$

where, $\mathbf{H}$ is the channel matrix that has a size of $N_r \times N_t$, $\mathbf{h}_{l_1}$ and $\mathbf{h}_{l_2}$are the l_1^{th} and l_2^{th} column vectors of $\mathbf{H}$, respectively and $\mathbf{n}$ is an additive white Gaussian noise vector with zero mean and a variance of σ^2. Moreover, the channel state information (CSI) is assumed to be fully known at the receiver side. Based on this formulation, we use the maximum likelihood (ML) detector, which is known to provide the optimum bit error rate (BER) performance for DSM. ML detector considers all potential realizations of the antenna indices, l_1 and l_2 and M-ary constellation symbols s_1 and s_2 to estimate $\tilde{l}_1$ and $\tilde{l}_2$ together with $\tilde{s}_1$ and $\tilde{s}_2$. This is achieved by searching over $N_t^2 M^2$ decision metrics and selecting the ones that satisfy the following cost function:

$$\left(\tilde{s}_1, \tilde{s}_2, \tilde{l}_1, \tilde{l}_2\right) = \arg \min_{s_1, s_2, l_1, l_2} \left\| \mathbf{y} - \left(\mathbf{h}_{l_1} s_1 + \mathbf{h}_{l_2} s_2 e^{j\theta}\right) \right\|^2 \tag{4}$$

3. Antenna Selection

Spatial modulation systems have many advantages, including: few radio frequency (RF) chains, ICI avoidance and low receiver complexity. However, accommodating transmit diversity into these systems is not straightforward [11]. One way to do that, though, is to use antenna selection (AS). The block diagram of TAS with DSM-MIMO systems is shown in Figure 3. Based on the channel estimation at the receiver, the best L_t out of N_t antennas are selected using one of the TAS schemes. After that, the DSM explained in Figure 2 is applied to the L_t transmit antennas instead of the total N_t transmit antennas.

To apply TAS, we select some ***columns*** from the channel matrix. The RF switch is controlled by the selection criteria implemented at the receiver. In TAS, the receiver feeds to the transmitter the L_t antenna indices to be utilized at each frame. Therefore, the received signal vector in (3) is modified to:

$$\mathbf{y} = \mathbf{H}_{Tx\_sel}\mathbf{s} + \mathbf{n} = \mathbf{h}_{\dot{l}_1} s_1 + \mathbf{h}_{\dot{l}_2} s_2 e^{j\theta} + \mathbf{n} \tag{5}$$

where, $\mathbf{H}_{sel}$ is the $N_r \times L_t$ modified channel matrix, $\dot{l}_1$ and $\dot{l}_2$ are the antenna indices chosen from L_t transmit antennas to transmit $s_1$and s_2, respectively and $1 \leq \dot{l}_i \leq L_t, i = 1,2$.

For the variety of MIMO techniques, the AS achieves the full diversity inherent in the system at the expense of a small loss in the coding gain in comparison to a full complexity system [17]. Many AS algorithms have been developed in the last decade. We can classify the AS algorithms into two main categories: (i) optimal AS algorithms, including EDAS which requires a high computational complexity at the receiver [14] and (ii) suboptimal AS algorithms which required a lower computational complexity at the receiver.

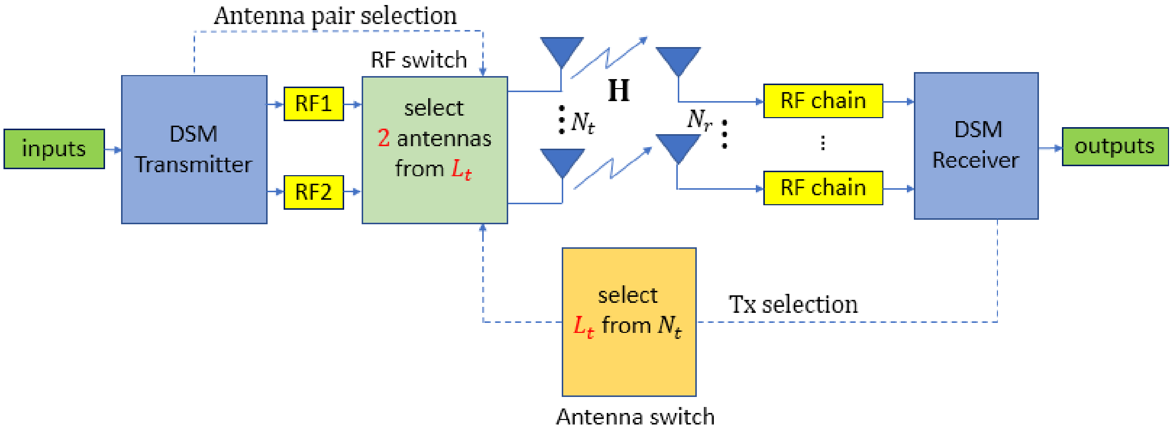


Figure 3. The block diagram of TAS for DSM scheme.

The optimal AS algorithm is the method that uses an exhaustive search of all possible combinations to find the one group that provides the best signal-to-noise ratio (SNR) for diversity or best capacity for spatial multiplexing. Therefore, the optimal AS algorithms require high computational processes at any change in the channel, which in turn leads to the difficulty of implementing these algorithms practically [18].

Since the optimal AS schemes suffer from practical limitations due to the high computational complexity, we will concentrate on using the sub-optimal AS algorithms and compare them with the optimal EDAS.

In this paper, we will focus on two suboptimal AS Algorithms for the DSM scheme. The first algorithm is capacity optimized AS (COAS) and the second is AS based on amplitude and antenna correlation (A-C-AS) [16]. Consider the channel matrix **H** has a size of $N_r \times N_t$. The best set of transmit antennas L_t are selected using one of the AS algorithms (COAS or A-C-AS) and the channel matrix. The selected transmit antennas L_t are used to convey the transmission vector **s** of the DSM.

3.1 Capacity Optimized Antenna Selection (COAS)

The COAS algorithm [16], also called norm-based antenna selection, is an AS algorithm that selects a sub-group of transmitting antennas (L_t) corresponding to the maximum channel amplitudes (columns of channel matrix) from the total number of transmit antennas N_t. The results of many research papers proved that the COAS algorithm was capable of enhancing the error performance of variety MIMO systems while imposing a very low computational complexity [16].

Transmit Antenna Selection (TAS) Based on COAS

The COAS algorithm can be applied as follows [16]:

Step 1: Calculate the Frobenius norm of each column **H**,

$$\|\mathbf{h}_i\|_F^2, \quad i = 1, 2, \ldots, N_t \tag{6}$$

Step 2: Re-arrange in descending order the columns,

$$\mathbf{H}_{\mathrm{A}} = \left[\|\mathbf{h}_1\|_F^2 \geq \|\mathbf{h}_2\|_F^2 \geq \cdots \geq \left\|\mathbf{h}_{N_t}\right\|_F^2\right] \tag{7}$$

Step 3: Choose the highest L_t channel gain vectors to form the $L_t \times N_r$ channel gain matrix $\mathbf{H}_{Tx\_sel}$.

3.2 Antenna Selection Based on Amplitude and Antenna Correlation (A-C-AS)

The A-C-AS algorithm is an AS algorithm based on the combination of two selection criteria: channel amplitude and antenna correlation. The correlation-based algorithm was introduced in [19]. TAS based on amplitude and antenna correlation (A-C-TAS) was first suggested for the SM by [16]. This scheme selects $L_t + 1$ transmit antennas that have the largest channel amplitudes from N_t total transmitting antennas. Thereafter, the correlations for all $\binom{L_t+1}{2}$ transmit antenna pairs are calculated. The transmit antenna pair that corresponds to the largest correlation is selected and the channel that has smaller channel gains within the selected pair is rejected. The A-C-AS scheme has shown a significant improvement in BER at low computational complexity. The smaller the correlation between transmitting antennas, the better the overall system performance [19].

Transmit Antenna Selection (TAS) Based on A-C (A-C-TAS)

The A-C-TAS algorithm can be applied as follows [19]:

Step 1: Calculate the Frobenius norm of each column vector in the channel matrix **H**,

$$\|\mathbf{h}_i\|_F^2, \quad i = 1, 2, \ldots, N_t \tag{8}$$

Step 2: Choose the $N_c = L_t + 1$ transmit antennas based on the largest norms of the column vectors,

$$\mathbf{H}_{N_c} = \left[\|\mathbf{h}_1\|_F^2 \geq \|\mathbf{h}_2\|_F^2 \geq \cdots \geq \left\|\mathbf{h}_{N_c}\right\|_F^2\right] \tag{9}$$

Step 3: Determine all possible enumerations of the channel gain vector pairs. The total number of possible vector pairs is given by $N_A = \binom{N_c}{2}$. Each pair will have the form (h_x, h_y).

Step 4: Calculate the angle of correlation θ between both vectors of a vector pair. For each vector pair, θ can be calculated as:

$$\theta_z = \cos^{-1}\left(\frac{|\mathbf{h}_x^H \mathbf{h}_y|}{\|\mathbf{h}_x\|_F\ \|\mathbf{h}_y\|_F}\right), \quad z = 1, 2, \dots, N_A. \tag{10}$$

The angle of correlation for each pair is stored in $\mathbf{A}_\theta$,

$$\mathbf{A}_\theta = \left[\theta_1\ \theta_2\ \dots\ \theta_{N_A}\right] \tag{11}$$

Step 5: Choose the largest correlation pair which has the smallest angle and reject the smaller of the two-channel gain vectors. This forms the $L_t \times N_r$ channel gain matrix $\mathbf{H}_{Tx\_sel}$.

3.3 Transmit Antenna Selection (TAS) Based on EDAS

Yang et al. [20] has introduced EDAS as the optimal AS for spatial modulation. The bit error performance of the SM scheme is improved significantly by maximizing the minimum Euclidian distance (ED) between all possible pairs of transmit antennas. The minimum ED for DSM is defined as:

$$ED_{min} = \underset{s_i, \hat{s}_i, l_i, \hat{l}_{i,}, i=1,2}{\arg\min} \left\|\mathbf{h}_{l_1}s_1 + \mathbf{h}_{l_2}s_2 e^{j\theta} - (\mathbf{h}_{\hat{l}_1}\hat{s}_1 + \mathbf{h}_{\hat{l}_2}\hat{s}_2 e^{j\theta})\right\|^2, \tag{12}$$

where, $\mathbf{v_1} = \mathbf{h}_{l_1}s_1 + \mathbf{h}_{l_2}s_2 e^{j\theta}$ and $\mathbf{v_2} = \mathbf{h}_{\hat{l}_1}\hat{s}_1 + \mathbf{h}_{\hat{l}_2}\hat{s}_2 e^{j\theta}$ is a pair of transmitted vectors and $\mathbf{v_1} \neq \mathbf{v_2}$. The EDAS can be applied on the DSM scheme by maximizing the Euclidian distance between all possible transmit vector pairs from the selected antenna sets. The L_t antennas that maximize ED_{min} in (12) are chosen for the EDAS transmission.

3.4 Computational Complexity for the AS Algorithms

We will evaluate the computational complexity for both AS algorithms in terms of the number of real multiplications (RM) and real addition (RA). Note that a complex multiplication is equivalent to 4 RM and 2 RA, ((a + jb) $*$ (c + jd) = ($a * c - b * d$) + j ($b * c + a * d$)), while a complex addition is equivalent to 2 RA, ((a + jb) + (c + jd) = (a + c) + j (b + d)) [21]. A similar approach of computational complexity analysis used in [15] is adopted in the following sub-sections.

Computational Complexity for COAS

The Frobenius norm in (6) needs N_r complex multiplication and $(N_r - 1)$ complex addition for each column vector in the channel matrix **H**. Then, the total number of real operations for each column equals, N_r (4 RM + 2 RA) + $(N_r - 1)$(2 RA) = 8 $N_r - 2$.
These operations are done N_t times. Therefore, the required number of real operations to compute is given by:

$$\mathcal{C}_{\text{COAS}-\text{TAS}} = N_t(8\,N_r - 2) \tag{13}$$

Computational Complexity for A-C-TAS

The Frobenius norm in (8) needs $N_t(8\,N_r - 2)$. The numerator in (10) requires N_r complex multiplication + $(N_r - 1)$ complex addition and 2 RM + 1 RA for evaluating the absolute value. So, the number of real operations for numerator equals N_r (4 RM + 2 RA) + $(N_r - 1)$(2 RA) + 2 RM + 1 RA = 8 $N_r + 1$.

In the denominator of (10), each Frobenius norm requires N_r complex multiplications, $(N_r - 1)$ complex additions and the multiplication of two Frobenius norms requires 1 RM. So, the number of real operations for denominator equals 2 (N_r (4 RM + 2 RA) + $(N_r - 1)$(2 RA)) + 1 RM = 16 $N_r - 3$.

Hence**,** the total number of real operations in (9) equals (8 N_r +1) + (16 N_r – 3) = (24 N_r – 2). These operations are done $\binom{N_c}{2}$ times. Therefore, the required number of real operations to compute ((8) and (10)) is given by:

$$\mathcal{C}_{\text{A}-\text{C}-\text{TAS}} = N_t(8\,N_r - 2) + \binom{N_c}{2}(24N_r - 2) \tag{14}$$

Computational Complexity for EDAS

The absolute value in Equation (11) and the summation of the resulted vector [1: N_r] need (2 RM+1 RA) N_r and RA ($N_r - 1$), respectively. So, the number of real operations for the absolute value equals (2 RM+2 RA) N_r - 1 RA= $4N_r - 1$.

The term $\mathbf{h}_{l_1}\mathrm{s}_1 + \mathbf{h}_{l_2}\mathrm{s}_2 e^{j\theta} - (\mathbf{h}_{\hat{l}_1}\hat{\mathrm{s}}_1 + \mathbf{h}_{\hat{l}_2}\hat{\mathrm{s}}_2 e^{j\theta})$ needs (4 complex multiplications, 1 complex addition and 2 complex subtractions) N_r. So, the number of real operations for this term equals (4(4 RM+2 RA)+2 RA+2(2 RA)) N_r =30 N_r.

An exhaustive search of (11) requires that the ED be calculated for all symbol combinations of s_i and $\hat{\mathrm{s}}_i$, such that $\mathrm{s}_i \neq \hat{\mathrm{s}}_i$. This requires $M^2(M^2 - 1)(34\ N_r - 1)$.

EDAS-DSM must then be done for each of the $\binom{N_t}{\mathrm{L}_t}$ transmit antenna subsets. Also, the EDAS-DSM must be performed for each antenna pair within each antenna subset; i.e., EDAS-DSM must be executed a total of $\binom{L_t}{2}\binom{N_t}{\mathrm{L}_t}$ times.

Therefore, the required number of real operations to compute (12) is given by,

$$\mathcal{C}_{\mathrm{ED-TAS}} = [(M^4 - M^2)(34\ N_r - 1)]\binom{L_t}{2}\binom{N_t}{\mathrm{L}_t} \tag{15}$$

3.5 Performance Analysis of DSM with TAS

The conditional pairwise error probability (PEP) for DSM is given by [10], [12]:

$$\mathrm{P}(x \rightarrow \hat{x}) = Q\left(d_{min}/\sigma\sqrt{2}\right) \tag{16}$$

where, $Q(.)$ is the tail distribution function of the standard normal distribution, $x = \mathbf{h}_{l_1}s_1 + \mathbf{h}_{l_2}s_2 e^{j\theta}$ is the transmitted vector which has been detected incorrectly as $\widehat{x} = \mathbf{h}_{\hat{l}_1}\hat{s}_1 + \mathbf{h}_{\hat{l}_2}\hat{s}_2 e^{j\theta}$ and d_{min} is the minimum Euclidian distance between all transmitted vectors.

In case of transmit antenna selection, $l_1, l_2, \hat{l}_1$ and $\hat{l}_2\ \epsilon[1, L_t]$ and the selected antenna set, L_t are chosen to maximize d_{min} in case of EDAS; whereas applying either COAS or A-C-TAS is expected to increase d_{min}. Therefore, considering (15), applying TAS leads to a decrease in the PEP for the same σ in TAS schemes.

4. SIMULATION RESULTS

The simulation result represents the average BER performance versus the average SNR at each receive antenna for different spectral efficiencies (4, 6 and 8 b/s/Hz). The optimum rotation angles for BPSK, 4-QAM are found as 90° and 45°, respectively [10].

All performance comparisons are measured at a BER equal to 10^{-5}. It has been assumed that all MATLAB simulations are performed over quasi-static Rayleigh fading channels. Additionally, it is assumed that the CSI is well known at receiver and the feedback link between the receiver and the transmitter is error-free. Furthermore, an optimal ML detection has been used at the receiver side.

Figure 4 shows the BER performance of the two sub-optimal TAS algorithms (COAS and A-C-AS) on DSM with spectral efficiency 4 b/s/Hz, different numbers of total transmit antennas N_t and the selected transmit antennas L_t equal 2. Both algorithms have been compared to each other.

The performance of COAS-DSM and A-C-AS-DSM schemes outperforms the conventional DSM (BPSK, $N_t = 2$) scheme with 1.5 dB and 4 dB, respectively when $N_t = 4$. However, this gain can be further improved by increasing N_t. Hence, when N_t is increased to 8, COAS-DSM and A-C-AS-DSM exhibit a 2.5 dB and 5.5 dB gains over the classical DSM. It is noted that by increasing N_t, the overall BER performance of AS scheme also increases. Furthermore, for a BER of 10^{-5}, A-C-AS-DSM outperforms COAS-DSM by 2.5 dB and 3 dB when $N_t = 4$ and 8, respectively. The optimal EDAS-DSM has a gain of 6.8 dB over DSM (BPSK, N_t = 2) . It outperforms the A-C-AS-DSM (BPSK, N_t = 8, L_t = 2) by 1.5 dB, but at the cost of high computational complexity.

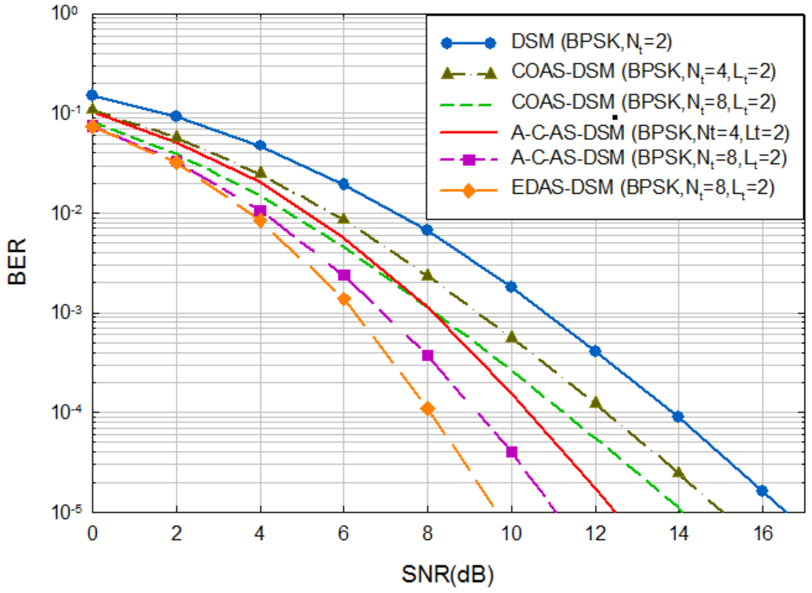


Figure 4. BER performance of TAS for DSM for 4 bits/s/Hz and $N_r = 4$.

Figure 5 illustrates the BER performance COAS and A-C-AS on DSM with spectral efficiency 6 b/s/Hz, different numbers of total transmit antennas N_t *and* the selected transmit antennas $L_t = 2$.

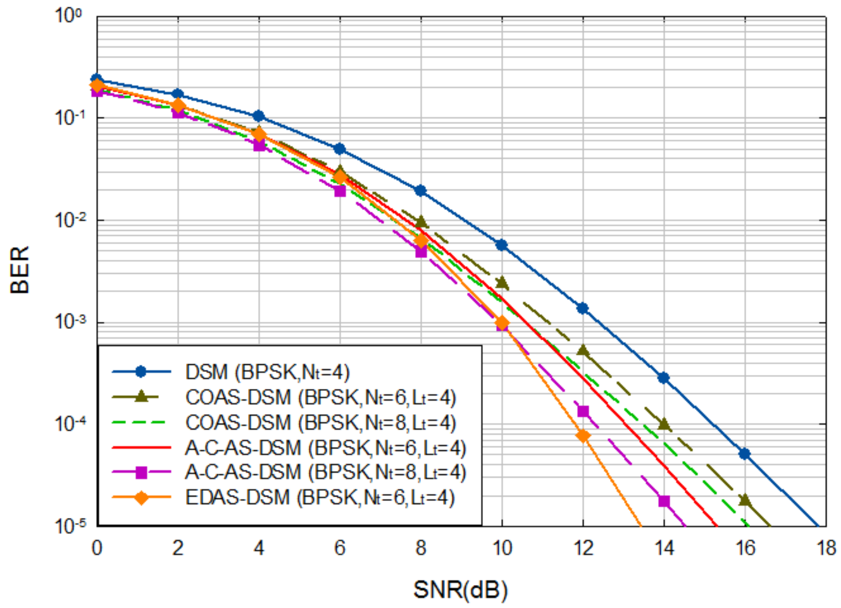


Figure 5. BER performance of TAS for DSM for 6 bits/s/Hz and $\boldsymbol{N_r} = \mathbf{4}$.

The performance of both COAS-DSM and A-C-AS-DSM schemes outperforms the conventional DSM (BPSK, $N_t = 4$) scheme with 1.2 dB and 2.5 dB, respectively when $N_t = 6$. However, this gain can be further improved by increasing N_t. For example, when N_t is increased to 8, COAS-DSM and A-C-AS-DSM exhibit gains of 1.8 dB and 3.3 dB over the conventional DSM.

It is noted that by increasing N_t, the overall BER performance of AS scheme improves. Furthermore,

for a BER of 10^{-5}, the A-C-AS-DSM outperforms COAS-DSM by 1.3 dB and 1.5 dB when $N_t = 6$ and 8, respectively.

Finally, EDAS-DSM has an estimated SNR gain of 4.2 dB over DSM for $N_t = 6$. İt outperform the A-C-AS-DSM (BPSK, Nt=6, Lt=4) by 1.9 dB. This comes again at the expense of higher complexity.

Figure 6 presents the results for a 8 × 4 16-QAM QSM and 4-QAM DSM systems with spectral efficiency 8 b/s/Hz, total transmit antennas $N_t = 8$ and the selected transmit antennas $L_t = 4$.

The COAS-DSM and A-C-AS-DSM schemes provide SNR gain of 0.6 dB and 1.8 dB over the conventional DSM (BPSK, $N_t = 8$), respectively. Accordingly, it can be seen that the COAS-DSM and A-C-AS-DSM schemes outperform the conventional DSM scheme, provided that both schemes have the identical spectral efficiency and the identical number of total transmit antennas. Furthermore, for a BER of 10^{-5}, the A-C-AS-DSM outperforms COAS-DSM by 1.2 dB.

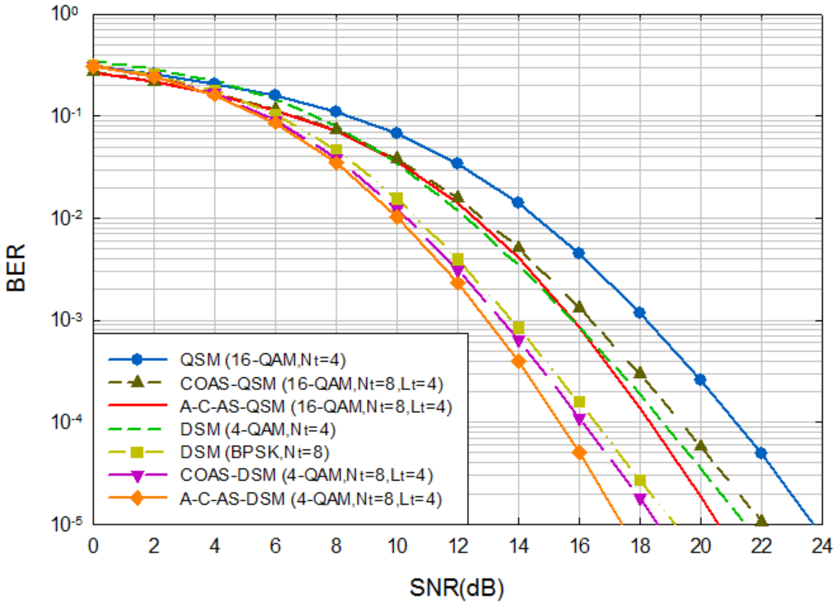


Figure 6. BER performance of TAS for DSM and QSM for 8 bits/s/Hz and $\boldsymbol{N_r = 4}$.

Also, the performance of COAS-DSM and A-C-AS-DSM schemes outperforms the conventional DSM (4-QAM, $N_t = 4$) scheme with 2.2 dB and 4 dB, respectively.

Finally, the conventional DSM (4-QAM, $N_t = 4$) outperforms the conventional QSM (16-QAM, N_t = 4) by 2 dB at 10^{-5}, while the gains for COAS-DSM and A-C-AS-DSM over COAS-QSM and A-C-AS-QSM are 3.3 dB and 3.2 dB, respectively.

The computational complexities for all the simulated cases are given in Table 1. It is clear that the complexity overhead for the antenna selection scheme is very small in case of the COAS scheme. The A-C-AS has a higher overhead which is reasonable with the achieved BER performance. However, the EDAS scheme has a very high computational complexity.

The overhead needed for antenna selection for both QSM and DSM is the same. The DSM has an increase in computation complexity compared to QSM, but it has a better BER performance.

5. Conclusions

In this paper, two sub-optimal antenna selection algorithms for DSM scheme were introduced. These algorithms are capable of improving the error performance of the DSM transmission scheme while requiring low computational complexities.

The COAS-DSM scheme has a lower computational complexity than the A-C-AS-DSM scheme, because it uses the channel amplitude as the selection criterion only. On the contrary, the A-C-AS-DSM scheme uses channel amplitude and antenna correlation as selection criteria. Therefore, the COAS-DSM gives a smaller improvement in BER performance than A-C-AS-DSM scheme. Still, there is a small loss of BER improvement of the A-C-AS-DSM compared with the optimal EDAS. This is justified by the much lower complexity needed for the A-C-AS-DSM. In other words, there is a trade-off between increasing computational complexity and improving error performance.

Table 1. Computational complexity for DSM and QSM with different antenna selection schemes.

Spectral efficiency	Transmission scheme	Mod. scheme	N_t	L_t	N_r	Real operation s	Comp. Overhead (%)
4	DSM	BPSK	2	2	4	1408	0
	COAS-DSM		4	2		1528	7.853403
	COAS-DSM		8	2		1648	14.56311
	A-C-AS-DSM		4	2		1810	22.20994
	A-C-AS-DSM		8	2		1930	27.04663
	EDAS-DSM		8	2		46768	96.98939
6	DSM	BPSK	4	4	4	5632	0
	COAS-DSM		6	4		5812	3.097041
	COAS-DSM		8	4		5872	4.087193
	A-C-AS-DSM		6	4		6094	7.581227
	A-C-AS-DSM		8	4		6154	8.482288
	EDAS-DSM		6	4		151432	96.28084
8	QSM	16-QAM	4	4	4	14336	0
	COAS-QSM		8	4		14576	1.646542
	A-C-AS-QSM		8	4		14858	3.513259
	DSM	BPSK	8	8		22528	0
	DSM	4-QAM	4	4		22528	0
	COAS-DSM		8	4		22768	1.054111
	A-C-AS-DSM		8	4		23050	2.264642

ACKNOWLEDGEMENTS

Thanks are due to Dr. Zohair Abu-Shahban for proofreading the article.

REFERENCES

[1] Goldsmith, Wireless Communications, Cambridge University Press, 2005.

[2] G. J. Foschini, "Layered Space-time Architecture for Wireless Communication in a Fading Environment when Using Multi-element Antennas," Bell Labs Technical Journal, vol. 1, no. 2, pp. 41-59, 1996.

[3] A. Elshokry and A. Abu-Hudrouss, "Performance Evaluation of MIMO Spatial Multiplexing Detection Techniques," Journal of Al Azhar University-Gaza (Natural Sciences), vol. 14, pp. 47-60, 2012.

[4] R. Y. Mesleh, H. Haas, S. Sinanovic, A. Chang Wook and Y. Sangboh, "Spatial Modulation," IEEE Transactions on Vehicular Technology, vol. 57, no. 4, pp. 2228-2241, 2008.

[5] M. Di Renzo, H. Haas, A. Ghrayeb, S. Sugiura and L. Hanzo, "Spatial Modulation for Generalized MIMO: Challenges, Opportunities and Implementation," Proc. IEEE, vol. 102, no. 1, pp. 56-103, 2014.

[6] E. Basar, M. Wen, R. Mesleh, M. Di Renzo, Y. Xiao and H. Haas, "Index Modulation Techniques for Next-generation Wireless Networks," IEEE Access, vol. 5, pp. 16693-16746, 2017.

[7] H. Zhang, L.-L. Yang and L. Hanzo, "Compressed Sensing Improves the Performance of Subcarrier Index-modulation-assisted OFDM," IEEE Access, vol. 4, pp. 7859-7873, 2016.

[8] H. Zhang, C. Jiang, L.-L. Yang, E. Basar and L. Hanzo, "Linear Precoded Index Modulation," IEEE Transactions on Communications, vol. 67, no. 1, pp. 350-363, 2018.

[9] R. Mesleh, S. S. Ikki and H. M. Aggoune, "Quadrature Spatial Modulation," IEEE Transactions on Vehicular Technology, vol. 64, no. 6, pp. 2738-2742, 2015, doi: 10.1109/TVT.2014.2344036.

[10] Z. Yigit and E. Basar, "Double Spatial Modulation: A High-rate Index Modulation Scheme for MIMO Systems," Proc. of IEEE International Symposium on Wireless Communication Systems (ISWCS, pp. 347-351, 2016.

[11] E. Basar, U. Aygolu, E. Panayirci and H. V. Poor, "Space-time Block Coded Spatial Modulation," IEEE Transactions on Communications, vol. 59, no. 3, pp. 823-832, 2011.

[12] Z. Sun, Y. Xiao, P. Yang, S. Li and W. Xiang, "Transmit Antenna Selection Schemes for Spatial Modulation Systems: Search Complexity Reduction and Large-scale MIMO Applications," IEEE Transactions on Vehicular Technology, vol. 66, no. 9, pp. 8010-8021, 2017.

[13] K. Ntontin, M. Di Renzo, A. I. Pérez-Neira and C. Verikoukis, "A Low-complexity Method for Antenna Selection in Spatial Modulation Systems," IEEE Comm. Letters, vol. 17, no. 12, pp. 2312-2315, 2013.

[14] S. Naidu, N. Pillay and H. Xu, "Transmit Antenna Selection Schemes for Quadrature Spatial Modulation," Wireless Personal Communications, vol. 99, no. 1, pp. 299-317, 2018.

[15] S. Kim, "Antenna Selection Schemes in Quadrature Spatial Modulation Systems," ETRI Journal, vol. 38, no. 4, pp. 606-611, 2016.

[16] N. Pillay and H. Xu, "Low-complexity Detection and Transmit Antenna Selection for Spatial Modulation," SAIEE Africa Research Journal, vol. 105, no. 1, pp. 4-12, 2014.

[17] G. Tsoulos, MIMO System Technology for Wireless Communications, CRC Press, 2006.

[18] A. F. Molisch, M. Z. Win, Y.-S. Choi and J. H. Winters, "Capacity of MIMO Systems with Antenna Selection," IEEE Transactions on Wireless Communications, vol. 4, no. 4, pp. 1759-1772, 2005.

[19] Z. Zhou, N. Ge and X. Lin, "Reduced-complexity Antenna Selection Schemes in Spatial Modulation," IEEE Communications Letters, vol. 18, no. 1, pp. 14-17, 2013.

[20] P. Yang, Y. Xiao, Y. Yu and S. Li, "Adaptive Spatial Modulation for Wireless MIMO Transmission Systems," IEEE Communications Letters, vol. 15, no. 6, pp. 602-604, 2011.

[21] R. Mesleh, O. Hiari, A. Younis and S. Alouneh, "Transmitter Design and Hardware Considerations for Different Space Modulation Techniques," IEEE Transactions on Wireless Communications, vol. 16, no. 11, pp. 7512-7522, 2017.

ملخص البحث:

التعديل الفضائي المزدوج (DSM) عبارة عن تقنية إرسال تم اقتراحها حديثاً لأنظمة الاتصال متعددة المداخل ومتعددة المخارج. وتمتلك هذه التقنية فاعلية طيفية أعلى مقارنة مع التعديل الفضائي الكلاسيكي؛ فهي تعمل على مضاعفة عدد هوائيات الإرسال الفعّالة.

في هذه الورقة، يتم تطبيق اختيار هوائيات الإرسال في التعديل الفضائي المزدوج من أجل تحسين الأداء من حيث معدّل خطأ البِت (BER). وبشكل أكثر تحديداً، نقوم بدمج خوارزميتين في التعديل الفضائي المزدوج؛ هما: اختيار الهوائيات القائم على السعة المثالية (COAS)، و اختيار الهوائيات بناءً على الاتساع والارتباط بين الهوائيات (A-C-AS). وقد تم عرض نتائج المحاكاة للخوارزميتين المذكورتين ومقارنتها مع طريقة اختيار الهوائيات الإقليدية المثالية المسندة الى المسافة المثالية (EDAS) باستخدام برمجية ماتلاب. وتبين النتائج وجود تسوية بين التعقيد والأداء. وعلى الرغم من وجود فقْد طفيف في معدّل خطأ البِت، فإن الخوارزميتين اللتين تم استخدامهما في هذه الدراسة أقل تعقيداً من خوارزمية (EDAS).

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 01, March 2020.

MODELLING MALWARE PROPAGATION ON THE INTERNET OF THINGS USING AN AGENT-BASED APPROACH ON COMPLEX NETWORKS

Karanja Evanson Mwangi[1], Shedden Masupe[2] and Jeffrey Mandu[3]

(Received: 10-Sep.-2019, Revised: 26-Oct.-2019, Accepted: 10-Nov.-2019)

ABSTRACT

Malware threat is a major hindrance to efficient information exchange on the Internet of Things (IoT). Modelling malware propagation is one of the most imperative applications aimed at understanding mechanisms for protecting the Internet of Things environment. Internet of Things can be realized using agent-based modelling over complex networks. In this paper, a malware propagation model using agent-based approach and deep-reinforcement learning on scale free network in IoT (SFIoT) is assiduously detailed. The proposed model is named based on transition states as Susceptible-Infected-Immuned-Recovered-Removed (SIIRR) that represents the states of nodes on large-scale complex networks. The reliability of each node is investigated using the Mean Time To Failure (MTTF). The factors considered for MTTF computations are: degree of a node, node mobility rate, node transmission rate and distance between two nodes computed using Euclidean distance. The results illustrate that the model is comparable to previous models on effects of malware propagation in terms of average energy consumption, average infections at time (t), node mobility and propagation speed.

KEYWORDS

Internet of things, Agent-based modelling and simulation, Modelling malware propagation, Large-scale-free networks, Deep-reinforcement learning.

1. INTRODUCTION

Today, any device connected to communication systems may be subject to unscrupulous and malicious individuals, whose main purpose is to access sensitive information. To achieve their goals, they use different specimens of malware [1]. This malware often goes unnoticed for a period long enough to study the behavior of the internal network and its elements, in order to extract valuable information. Considering that there are large numbers of nodes deployed on communication systems and, in many cases, they are usually deployed on hostile unattended environments without human supervision, they become a principal target for malware attacks [2]. Agent-based modelling and simulation (ABMS) is an effective way to model and analyze complex networks [3]. Network consists of agents and the activities of these agents are monitored concurrently. ABMS offers the set of transition rules with consideration to individual device characteristics thus appropriate for malware modelling, where individual device variability is a key consideration [4]-[5]. This paper postulates the malware propagation process on scale-free networks by proposing agent-based model and simulation. In scale-free networks, nodes are added with maximum probability node. Agent-based modelling and simulation are instigated for modelling the dynamics of malware propagation scale-free networks. The diversity of nodes in scale-free network by varying parameters, such as node mobility, energy consumption and propagation speed that affect the malware spread in the network. The proposed model is further compared with analytical results obtained from previous agent-based modelling and simulation schemes [6]–[9]. The major contributions of this paper are outlined as follows:

1) Creation of an agent-based model and simulation with a decision maker for modelling the malware propagation on large networks using a deep-reinforcement learning algorithm.
2) The node state transition model Susceptible-Infected-Immuned-Recovered-Removed (SIIRR) is developed and the individual node performance measurement is estimated for computing the node reliability using mean-time-to-failure metric.

[1] K. E. Mwangi is with Faculty of Engineering, University of Botswana Gaborone, Botswana. Email: sun-dayfeb29@gmail.com

[2] S. Masupe is with BITRI, Botswana. Email: smasupe@bitri.co.bw

[3] J. Mandu is with Department of Electrical Engineering, University of Botswana Gaborone, Botswana. Email: jef-freym@mopipi.ub.bw

The rest of the paper is structured as follows: The related literature on malware propagation is explored in section 2. In section 3, the proposed model is presented and the succinct details on the application of deep-reinforcement learning in modelling malware propagation are given. The experimental set-up and simulation of the proposed scheme are discussed in section 4. Analysis is performed to compute the metrics, such as average energy consumption, average infections over time, node mobility and propagation speed. The simulation results are validated and compared with analytical results obtained from previous agent-based modelling and simulation schemes. Finally, the conclusion of this paper and future research directions are given in section 5.

2. Related Literature

The rise in use of IoT devices to launch malware attacks in the recent past has invoked researchers' interest in understanding IoT malware propagation and control. In this section, we review recent literature in malware propagation with a bias towards agent-based modelling which is the approach taken in the our proposed model.

A Markov Random Filed (MRF)-based spatio-stochastic framework is applied in complex communication networks, where malicious threats spread through direct interactions and follow the SI state model proposed by Karyotis [8]. It also combines Gibbs sampling with simulated annealing to analyze the behaviour of the systems under various topological and malware-related metrics. The disadvantage of MRF is that it is not isotropic, since it varies in magnitude according to the direction of measurement. Besides, the reliability of individual nodes is not assessed. The rumor diffusion process is proposed to model the outbreak of malware in [7]. The limitation of this agent-based analytical model is that it is difficult to prove the validity of the malware-free equilibrium stability (global and local).

In [10], the four aspects of malware propagation modelled were; user mobility, application-level interactions among users, local network structure and network coordination of malware (Botnets). The model was tested for a malicious virus like Cabir spreading among the cellular network subscribers using Bluetooth. A queuing-based malware propagation modelling approach was proposed in complex networks with churn [11]. Churn refers to dynamic node variation which captures the dynamics of SIS-type malware in time- varying networks. It quantified network reliability and improved the robustness of the network against some generic malware attacks. With the dynamic nature of node variation, it does not consume less energy and also the spreading speed is high. Malware propagation over wireless sensor networks has been proposed in [12], where the network topologies are based on complete or regular graphs. The first disadvantage of this network model is that it does not consider the individual characteristics of sensor nodes which form an important attribute in modelling heterogeneity of nodes and the second disadvantage involved in this model is that parameters such as transmission rate and recovery rate are not explicitly defined.

Batool et al. [9] demonstrates that Internet of Things networks can be modelled using a hybrid approach of using complex network and agent-based models. The construction of IoT elaborated models addressing the emergence and individual characteristics represent an existing research challenge. To model the IoT as a scale-free network, when a new node wants to join the network, it requires the degree and distance of all nodes (centrality measures) in the whole network in order to compute the probability of connecting to each existing node. The centrality measure is a critical measure of how central the node is to communication and connectivity. Betweenness and closeness centralities are calculated in each subnet. Betweenness centrality of a node is the probability for the shortest path between two randomly selected nodes to go through that node and is calculated as:

$$C_B(i) = \frac{1}{(n-1)(n-2)} \sum_{j \neq i, k \neq i, j \neq k} \frac{N_{sp}(j \overset{i}{\rightarrow} k)}{N_{sp}(j \rightarrow k)} \tag{1}$$

where, $Nsp(j \rightarrow k)$ is the number of shortest paths from node i to node k and $Nsp(j \xrightarrow{i} k)$ is the number of shortest paths from j to node k that pass through i.

Closeness centrality is a measure of how accessible a node is from other nodes and is calculated as:

$$C_c(i) = (\frac{\sum_j d(i \rightarrow j)}{n-1})^{-1} \tag{2}$$

which is an inverse of the average distance from node *i* to all other nodes. If *Cc*(*i*) =1, then you can reach each other node in the network *via* one step. The centrality measures are key to determine the influence of malware propagative and spreading nodes.

The inherent weakness of the deterministic and stochastic models surveyed in our previous work in literature is the full mix assumption [13]. The full mix assumption holds that every node has equal chances of coming into contact with others in the network, which is not necessarily the case in malware propagation on IoT networks where heterogeneity is a key factor. The introduction of the decision maker in the model overcomes the key challenge of arriving at an infection decision based on individual node interaction and individual node parameters, not just contact.

3. The Proposed Model

In this section, a model is formulated to model malware propagation over large-scale-free communication networks. A scale-free network environment for heterogeneous IoT devices is visually illustrated in Figure 1.

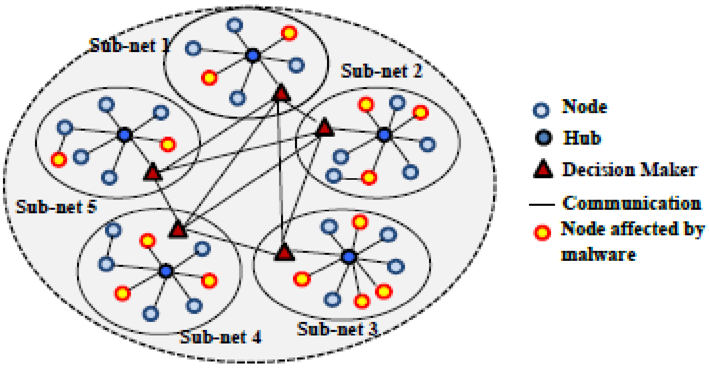


Figure 1. Scale-free Internet of Things networks.

The notion for modelling of malware propagation on large-scale-free networks is as follows; mitigate effects of malware over large-scale-free IoT networks, set flexible simulation parameters (number of nodes/devices are high and transmission range is also high), reduce the malware propagation speed in SFIoT networks and analyze regular changes in the subnets due to the node mobility rate between subnets within a time-varying environment. We consider a network as a graph with *N* nodes and *M* edges. The total population of *N* nodes is divided into *T* subnets, with n_i nodes where i=1,2,. . . , m nodes. The total population of nodes is given by Equation 3:

$$\sum_{i=1}^{m} n_i = N \tag{3}$$

For each subnet *T*, the probability P_i is used to add a link between two nodes that should satisfy Equation 4:

$$\sum_{i=1}^{m} n_i P_i . \frac{1}{2} n_i (n_i - 1) = \frac{N(K)}{2} \tag{4}$$

where, K denotes average degree of nodes in the entire network. When a new node is announced to the network to be attached to N nodes with high degree K, the announcement of the new node and preferential attachment continue until a network with !=t+N has been deployed. The principle of the decision maker-based model of malware propagation on sub-netting-based scale-free networks is based on the SIIRR model states. Decision maker is denoted as an agent considered for modelling malware propagation. Each node in the network has defined heterogeneity behaviour and set of rules is used for modelling the node behaviour. While modelling the malware propagation, nodes are classified into five states. In each time stamp, a node transits to one of the five possible states as listed below.The state transition diagram for SIIRR model is depicted in Figure 2.

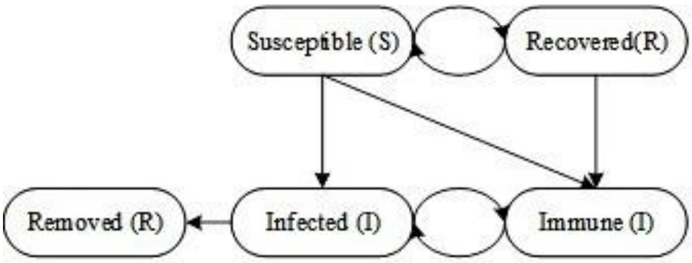


Figure 2. SIIRR model transition diagram.

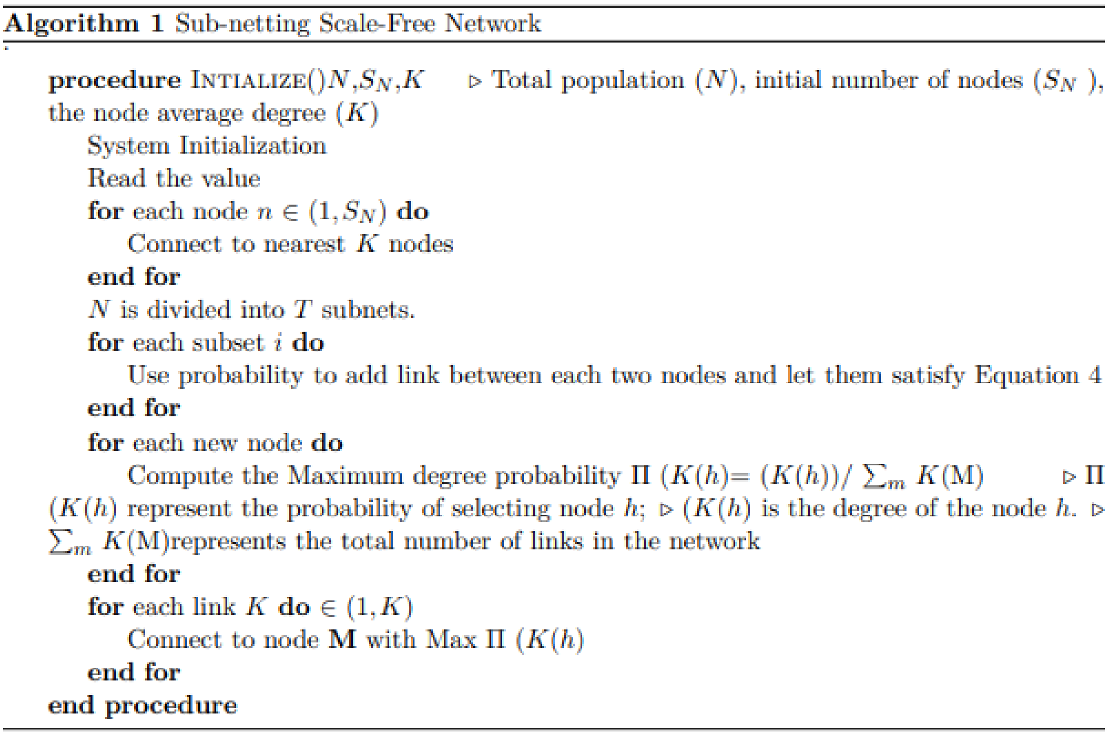
1) Susceptible (S): It is the first state of a node or hub and it often refers to infected in future.

2) Infected (I): The node attracted by malware is called the infected node. In this state, a node propagates the malware infections to all their neighbours.

3) Immune (I): The node that is unable to become infected by any node is called immune. This type of nodes has an immunization scheme, such as an anti-malware solution, to detect and block malware.

4) Recovered (R): It refers to infection removed state and does not get infected again.

5) Removed (R): The node or hub is attracted by the malware and can spread malware at time, *t*.



Algorithm 1 Sub-netting Scale-Free Network

```
procedure INTIALIZE()$N$,$S_N$,$K$    ▷ Total population ($N$), initial number of nodes ($S_N$ ), the node average degree ($K$)
    System Initialization
    Read the value
    for each node $n \in (1, S_N)$ do
        Connect to nearest $K$ nodes
    end for
    $N$ is divided into $T$ subnets.
    for each subset $i$ do
        Use probability to add link between each two nodes and let them satisfy Equation 4
    end for
    for each new node do
        Compute the Maximum degree probability $\Pi$ ($K(h)$= ($K(h)$)/ $\sum_m K$(M)    ▷ $\Pi$ ($K(h)$ represent the probability of selecting node $h$; ▷ ($K(h)$ is the degree of the node $h$. ▷ $\sum_m K$(M)represents the total number of links in the network
    end for
    for each link $K$ do $\in (1, K)$
        Connect to node M with Max $\Pi$ ($K(h)$
    end for
end procedure
```

The flowchart in Figure 3 shows the steps in model formulation. Algorithm 1 shows the detailed procedure for sub-netting-based network construction.

3.1 Modelling Deep-reinforcement Learning in Malware Propagation

A Deep-reinforcement Learning (DRL) scheme is adopted to illustrate the variables used for a Continuous Markov Chain Model (CMCM). The main goal of the CMCM in a DRL problem is to increase the obtained rewards. The tuples of DRL are as follows:

$$T = S, A, R, E, H, \gamma \tag{5}$$

where, S denotes the set of states, A is a possible set of actions, E is the environment, R is the reward function for state and action. In DRL, the agent has the ability to act where each action influences its

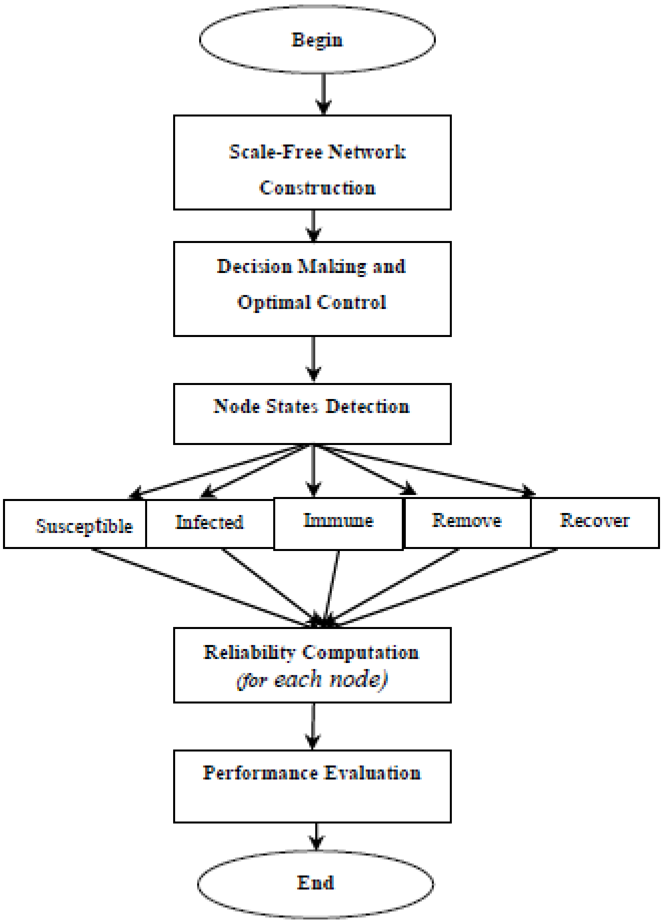

Begin
Scale-Free Network Construction
Decision Making and Optimal Control
Node States Detection
Susceptible
Infected
Immune
Remove
Recover
Reliability Computation (for each node)
Performance Evaluation
End

Figure 3. Scale-free Internet of Things (SFIoT) malware propagation model.

future state of the agent and success can be estimated using scalar reward signal. Q-learning-based reinforcement learning algorithm solves the decision making problems. Q-learning is defined as the quality of action in given state S at time t.

Environment (E): The environment is the area in which agents communicates with each other.
Agents (A): In a given environment, an agent receives information and performs the corresponding action. The main goal of agents is to pick the best policy that increases the total reward.
States (S): This is the condition defined by agent characteristics within the defined transitions.
Actions (A): A state transition from one state S_t to another state $S_{(t+1)}$at time t+1 is called action.
Reward (R): It represents the closeness of the current state to the true class. It is formulated by Equation 6.

$$R\big(S_t A_t S_{(t+1)}\ Y\big) = C\ \big(S_{(t+1)}, Y\big) \tag{6}$$

Rewards depend on the current state and the action performed.
Discount factor (γ): The discount factor controls the importance of future rewards ($\gamma \in [0,1]$).
State transition distribution: It is the transition probability that action A in state S at time t will lead to state S^t at time $t+1$: $PA(S,S^t)=PR(S^t\,|S,A)$. The policy (π) where (π)= At and the policy for a state is denoted $(\pi)(S) \longrightarrow A$ which changes with the reward policy as:

$$\Re_t = \sum_{t=0} \gamma^t\ R_t\,\gamma \tag{7}$$

where $0 \leq \gamma < 1$.

In the Q-learning approach, an approximate reinforcement machine learning algorithm is presented for IoT devices. Consider the Q-value updated equation as formulated in Equation 8.

$$Q(S_{t+1}, A_{t+1}) \Leftarrow (1-\alpha)Q(S_t, A_t) + \ \alpha\ [\Re\ (S_t, A_t) + \gamma\ \max_{a'} Q(P(S_t, A_t), a')] \tag{8}$$

where, $Q(S_t,A_t)$ is the Q-value of current state S_t when action A_t is selected at time t, α is the learning rate, γ is the discount factor, where γ is set between 0 and 1, $\max_{a'} Q(P(S_t, A_t), a')$ is the maximum possible Q-value in the next state $S(\ +1)$ if selects possible action a'. $\Re\ (S_t, A_t)$ denotes the reward function when state St selects At.

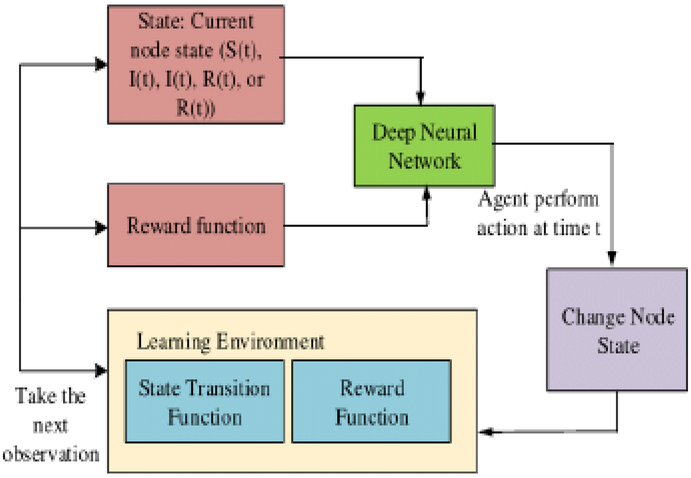

State: Current node state (S(t), I(t), I(t), R(t), or R(t))
Deep Neural Network
Reward function
Agent perform action at time t
Change Node State
Learning Environment
State Transition Function
Reward Function
Take the next observation

Figure 4. Deep-reinforcement learning.

Figure 4 visually illustrates the deep-reinforcement learning approach adopted in the model. The Q-learning model is used to classify the nodes as part of five possible transition states. It specifies transition of nodes between states from $S \rightarrow I$, $I \rightarrow R$ and $R \rightarrow S$, where the recovered state and removed state are terminal. The nodes do not transition to another state after being in the removed state or the recovered state. It is represented as the SIIRR model and mathematically formulated as:

$$\frac{dS(t)}{dt} = -\sigma \frac{S(t)\, I(t)}{N} \tag{9}$$

$$\frac{dI(t)}{dt} = -\sigma \frac{S(t)\, I(t)}{N} - \alpha\, I(t) \tag{10}$$

$$\frac{dI(t)}{dt} = \beta\, I(t) \tag{11}$$

$$\frac{dR(t)}{dt} = \alpha\, I(t) \tag{12}$$

$$\frac{dR(t)}{dt} = \sigma\, I(t) \tag{13}$$

where, _ is the infection rate S ! I, _ is the recovery rate I ! R, _ is the removed rate. The total population N (network size) at time t is computed as:

$$N(t) = S(t) + I(t) + I(t) + R(t) + (R(t) \tag{14}$$

After the scale-free network formation, all the hubs, decision makers and ordinary nodes are set to susceptible state. At time slot t = 1, one or more nodes are set into infected state and each time slot t = 2, 3 or 4 . . . n, malware propagates from infected nodes to their adjacent nodes through communication links. The node state changes continuously at each time slot.

3.1.1 Reliability Computation

The reliability function for a node is computed by using Mean Time To Failure (MTTF). However, most of the previous schemes in malware modelling have not considered the reliability factor. Specifically, reliability is the probability that the system will perform its intended function according to the specified design. To improve the network performance, we consider several metrics for computing the reliability. These are; node degree, node mobility rate, node transmission rate and distance between two nodes.

Node degree is the number of links (in degree and out degree) that lead into or out of the node. For each sub-net, the mobility of the node (i) is computed as follows:

$$M\ (i) = \sum_{i=1}^{n} \frac{NCP - NOP}{Mobility\ Speed} \tag{15}$$

where NCP is the Node Current Position and NOP is the Node Origin Position. A transmission rate (in Kbps) between two nodes depends on the message size (D_s) and distance between two nodes (DN) given as:

$$T\ R(i) = C_1 \times D_S + C_2\ DN \tag{16}$$

where C_1 and C_2 are constant variables. The distance between two nodes is computed using the Euclidean distance metric, which is calculated as:

$$d(a,b)^2 = (b_1 - a_1)^2 + (b_2 - a_2)^2 \tag{17}$$

The reliability of a node $R(N(t))$ is the probability that the node will be successful in the interval between time 0 and t as shown in Equation 18:

$$R\big(N(t)\big) = P(r > t) \quad t \geq 0 \tag{18}$$

In Equation 18, r is a random variable that denotes the time-to-failure or failure time. The mean time to failure is computed by Equation 19:

$$MTTF = \int_0^\infty t\ f(t)dt, Then\ f(t) = -\frac{dx}{dt}[R(t)] \tag{19}$$

Performing integration operation yields;

$$MTTF = \int_0^\infty td\ [R\ (N(t))] = \int_0^\infty t\ [R\ (N(t))]\ + \int_0^\infty R\ \big(N(t)\big)dt \tag{20}$$

In Equation 20, t(R(N (t) → 0 and x → ∞. It yields the second term, which equals:

$$MTTF = \int_0^\infty R\ \big(N(t)\big)\ dt \tag{21}$$

For each sub-net in a scale-free network, the reliability of a sub-net at time t can be computed by:

$$R\big(S(t)\big) = 1 - \prod\nolimits_{sub-net \in path} (1 - R\big(N(t)\big) \tag{22}$$

Moreover, any path composed of sub-nets in a scale-free network R(P(t)) at time t can be computed as:

$$R\big(P(t)\big) = \prod\nolimits_{sub-net \in path} (1 - R\big(N(t)\big) \tag{23}$$

As a result, a sub-net-based scale-free network consists of reliable paths. Hence, the reliability of the network (R(t)) is computed by;

$$R(t) = 1 - \prod\nolimits_{path} (1 - R\big(P(t)\big) \tag{24}$$

The topology of a scale-free network is constructed based on the actual parameters (node degree and maximum probability of a node) in a sub-net. The proposed scheme is implemented in the field of Internet of Things. The reliability for each node in the scale-free network is under malware propagation situation.

4. Simulation

In this section, the modelled propagation algorithm is simulated. The proposed scheme was compared to analytical results obtained from published works as follows: for energy consumption, to the work of Batool et al. [9]; for average infection rate, to the works of [6], [7]-[14]; for propagation speed and node mobility to the work of [8] based on the performance metrics described in sub-section 4.2.

4.1 Experimental Set-up

The model is implemented using NS-3 (version 3.26) for simulation. NS-3 is a network simulator which is mainly supported for Linux and written using C++. But, the binding of NS-3 is written in Python. In our experiment, the Gaussian Markov (GM) mobility model is used. Gauss-Markov (GM) mobility model is used to simulate mobility of device agents. Gauss-Markov mobility model caters for temporal dependency; i.e., it has a memory to correlate previous states. In Gauss-Markov, the velocity of the device is modeled as a Gauss- Markov stochastic process, as it is assumed to be correlated over time. In this model, node speed and direction are considered with respect to time, taking into account the previous speed s_{n-1}, previous direction d_{n-1}, the mean speed $\bar{s}$ and direction $\bar{d}$. The randomness parameter α has a Gaussian distribution. Current speed and direction are given by:

$$s_t = \alpha\, s_{t-1} + (1-\alpha)\,\bar{s} + \sqrt{(1-\alpha^2) s_{xn-1}}$$

$$d_t = \alpha\, d_{t-1} + (1-\alpha)\,\bar{d} + \sqrt{(1-\alpha^2) ds_{xn-1}} \qquad (25)$$

where, s_{xn-1} and d_{xn-1} are random variables from a Gaussian distribution. The simulation of the proposed scheme uses 200 node moves in a 5000 m × 5000 m rectangular region for 100 seconds of simulation. These nodes are vehicles deployed along the road perimeters and 20 sensors are used for sensing information.

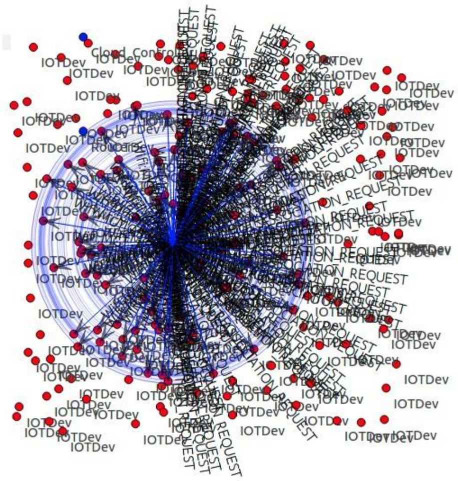


Figure 5. Scale-free network formation visualization.

Four traffic lights for each road lane entering the intersection are considered. The blue circle in the upper right section represents the decision maker entity that manages the traffic light timing. Assume that each node moves independently with the same average speed. All nodes in the network have the same transmission range of 250 m. The simulated traffic is of a Constant Bit Rate (CBR). The proposed scheme is implemented in a single intersection-based road traffic system, then the sub-net construction process is performed. The process is based on the node residual energy and degree of nearest node. In each sub-net, decision maker is selected. All nodes are connected into hub. If the node is not connected

Table 1. Simulation settings and parameters.

Simulation parameters	**Values**
Network simulator	NS-3.26
Area size	5000 m×5000 m
No.of nodes	200
Communication range	250m
Simulation time	100 seconds
Packet size	1024 bytes
Mobility model	Gauss-Markov Model
Node speed	2, 4, 6, 8 and 10 m/s
Pause time	5 seconds
No. of runs	100
No. of packets	100 packets /simulation

to hub, the route between the node and hub is found using FIFO rule. Next, the node sense data and decision maker classify the node state as susceptible, infected, immune, recovered and removed using Deep-Reinforcement Learning (DRL). A visualization showing the formation of a scale-free network can be seen in Figure 5. The simulation settings and parameters are summarized in Table 1.

4.2 Simulation Performance Metrics

The proposed scheme is evaluated for performance based on the following metrics:

1) Energy consumption: It is the rate of energy used for packet transmission. Energy conservation is an important issue while communicating with other nodes.
2) Average infection rate: It is the number of nodes found to be infected during packet transmission.
3) Propagation speed: It can be computed by finding the number of infected nodes at time *t* and is based on the threshold value for different states.
4) Node mobility: It has long been recognized as an efficient metric for modeling malware propagation in Internet of Things; e.g. road traffic systems and smart office application systems. It causes major issues, such as increased energy consumption and connectivity failure. Hence, it needs to be considered in complex networks, so that it brings benefits of reduced energy consumption and reduced spread of malware over communication networks.

4.3 Comparative Analysis

The statistical analysis of the obtained simulation raw data is carried out. Average (means) and the confidence intervals are calculated. The confidence interval of the data realized from the simulation is calculated as follows. Simulations $x1,x2,...,x5$ are carried out for each set of network size in the simulation. Since the number of sample simulations is less than 30, that is *n* = 5, the *t* distribution with n-1 degrees of freedom is adopted as the statistical test. In order for the the *t* distribution to be applied, the data needs to follow normal distribution. The test for normality is carried out to provide evidence that the simulation data is normally distributed. The normal probability plots are used to depict the outcome of the normality test. Shapiro-Wilk normality measure is also applied, since simulation instances are less than 2000. Shapiro-Wilk test is carried out at all network sizes. The confidence interval is given as [L, U], where L is the lower bound and U is the upper bound of the interval. This can be expressed as [L, U] = [average – margin of error, average + margin of error]. The confidence interval is calculated as:

$$[L,U] = \left[\bar{x} - t_c\frac{s}{\sqrt{n}}, \bar{x} + t_c\frac{s}{\sqrt{n}}\right] \tag{26}$$

where, tc is the critical value from the *t* distribution depending on the confidence level. The confidence level of 95% is used in this study.

The simulation results are subject to the test of normality for each of the network sizes and parameters. Shapiro-Wilk test statistics and the normal probability plots are derived for each of the network sizes and parameters. The normal probability plot is a visual illustration showing whether the data fits a normal probability distribution. The simulation raw results are plotted against the theoretical quantiles. If the data lies along the straight, that data fits the normal probability distribution. The test proved that the results on all network sizes were normally distributed as required for the use of Student *t* distribution in the calculation of confidence interval. For illustration purposes, the example of the normal probability plots for network size of 60 nodes is shown here. Figure 6 shows the normal probability plots for a network of 60 nodes.

Shapiro-Wilk test statistics are calculated based on the following hypotheses:

H0: The population is normally distributed.
H1: The population is not normally distributed.

If the significance level Sig. = α > 0.05, we can’t reject H0, thus the population is normally distributed. Shapiro-Wilk test statistics indicate that all the data from the simulations is normally distributed at 95% confidence interval.For example, the network size of 60 nodes shown in Figure 6 yielded Shapiro-Wilk test statistics and confidence levels as shown in Table 2.

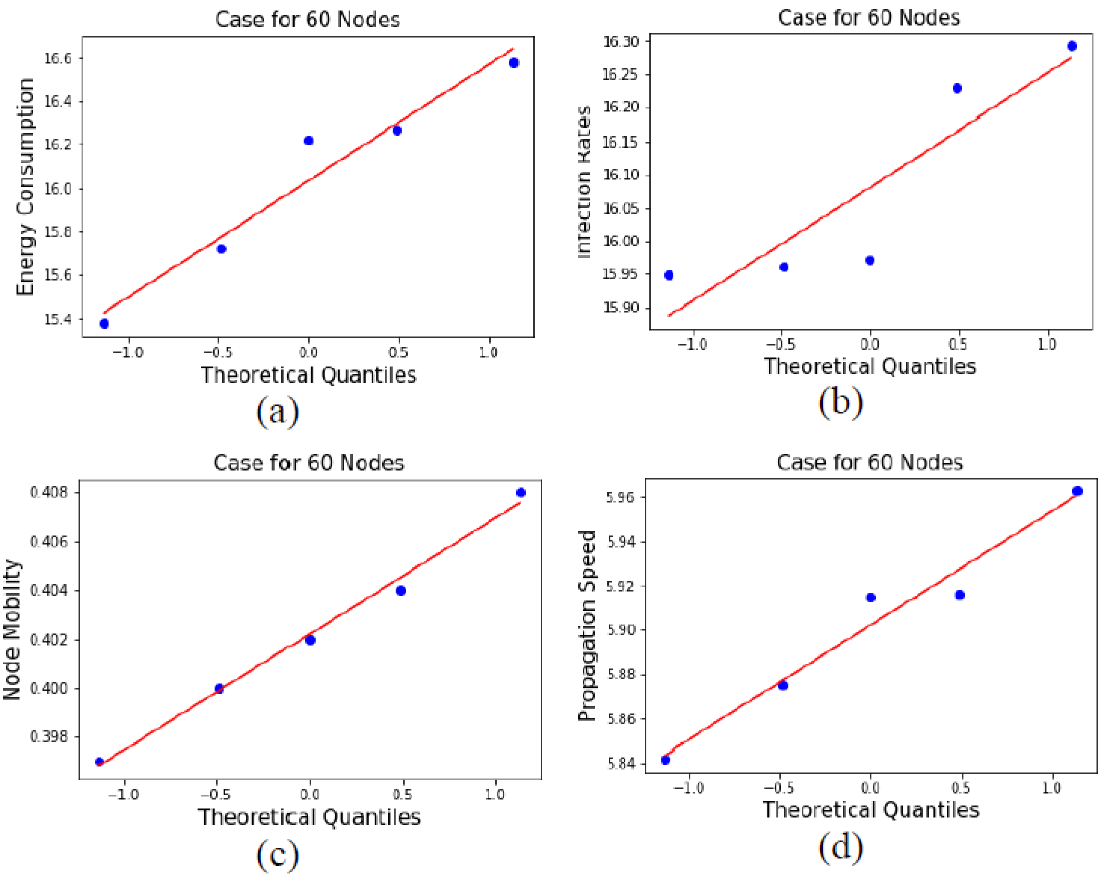


Figure 6. Normal probability plots for network size = 60 nodes for (a) Energy consumption (b) Infection rates (c) Node mobility and (d) Propagation speed.

From Table 2, the significance level **Sig.** = $\alpha > 0.05$ satisfies $H0$ and the data is normally distributed. The 95% confidence level upper and lower bounds are also calculated.

Table 2. Test on network size of 60 nodes.

	Shapiro-Wilk test significant levels	Mean Difference	95% confidence level of the difference	
	If (Sig.>0.05), Accept H0		Upper (U)	Lower (L)
Propagation	0.871	5.902200	5.8525	5.95915
Energy	0.706	16.034200	15.44065	16.62775
Mobility	0.995	0.40220	0.39705	0.40735
Infection Rate	0.55	16.080600	1544065	16.2865

4.3.1 Energy Consumption

First, we examine the energy consumption for our proposed scheme and then compare with the previous scheme. Energy consumption is the practice of quantity of energy used. It can be achieved through efficient energy use over complex communication environment. The tasks that are considered for energy consumption include: sensing, transmission and communication. The total energy consumption was estimated in milli joule (mJ). It is formulated as follows:

$$E_c = E_T + E_R + E_I \tag{27}$$

Energy consumption for transmission, E_T is computed by:

$$E_T = (\alpha_1 + \alpha_2 D^{\sigma})m \tag{28}$$

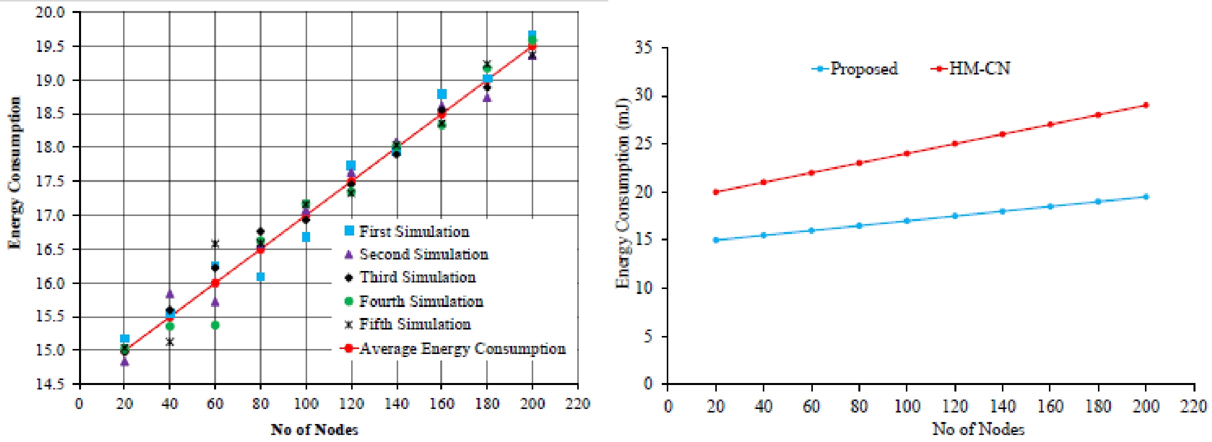
Energy consumption for reception E_R is computed by:

$$E_R = (\alpha_3)\, m \tag{29}$$

Energy consumption for idle state E_I is computed by:

$$E_I = \alpha_4 t_I\, P_m \tag{30}$$

In Equations from (27) to (30), D is the transmission distance, m is the packet length, α_1 - α_4 are the system dependent parameters, t_I is the idle time and P_m is the packet processing rate of the node. Five simulations were carried out for each network size and energy consumption measurements were noted for each run. Figure 7 shows the average energy consumption comparative analysis. Sub-Figure 7(a) shows the energy consumption rates at varied network sizes on the proposed scheme and in Sub-Figure 7(b), the average rate of energy consumption for the proposed scheme and that of HM-CN [6] are compared.



((a)) Simulated Energy Consumption.

((b)) Average Energy for the proposed model Vs HM-CN [6].

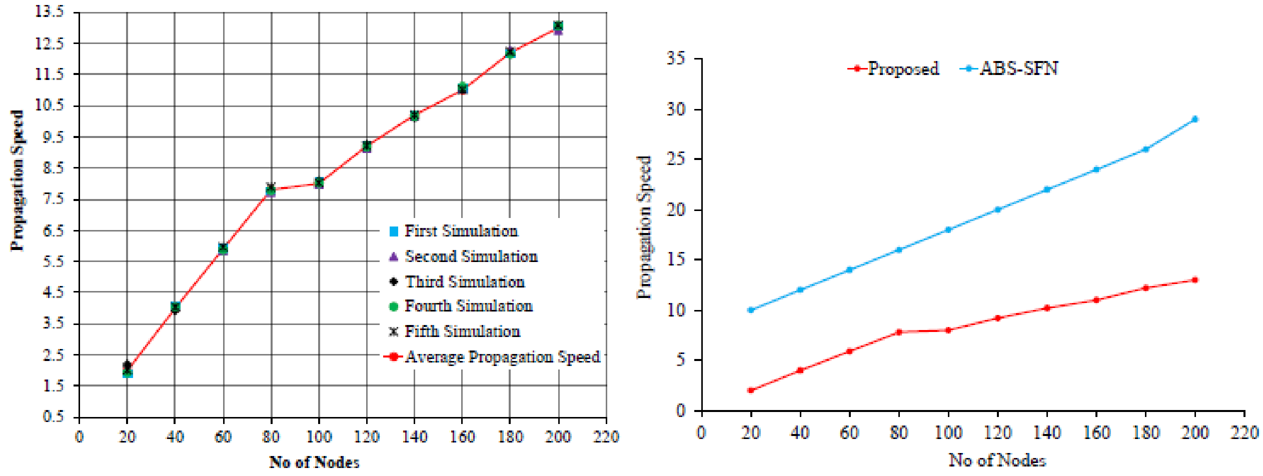
Figure 7. Energy consumption analysis.

Previous work; namely, HM-CN [6], noted that sensing and communication are the most energy-consuming tasks. Transmission and reception cost is high, especially for short-range communication. These drawbacks are solved and our proposed scheme provides a realistic estimation of energy consumption in networks. The proposed scheme is simulated for N=200 nodes (nodes varying as 20, 40,...,200). The decision maker isolated malware-infected nodes which are not allowed for communication and sensing. Furthermore, we follow FIFO rule for packet transmission. Hence, we obtained minimum energy consumption.

4.3.2 Average Infection Rate

Infection rate is an important parameter in modelling malware dynamics and propagation. During malware behaviour modelling, there is a need to examine the effect of the infection rate of each node and compute the average infection rate for various network sizes. Simulations were taken for network size variations. Figure 8(a) illustrates the infection rates at varied network sizes. The proposed scheme infection rates are based on the scale (threshold) of malware prevalence and the scheme is compared to the scheme with Dynamic Analysis and Control (DAC) scheme [6], Rumour Spreading Process-Scale Free Networks (RSP-SFN) [14] and Markov Random Field-Complex Communication Networks (MRF-CCN) [8]. A snapshot of the proposed *vs.* previous schemes in terms of infection rate is depicted in Figure 8(b).

From the simulation results, the proposed scheme gave less number of infections per given number of nodes. The threshold of α is directly proportional to the malware infections. If α is small, the number of infected hosts will largely increase. In Dynamic Analysis and Control (DAC) [6], the propagation control strategies did not perform well, hence decreasing the real-time immunity rate and increasing the proportion of infected nodes. In Rumour Spreading Process-Scale Free Networks (RSP-SFN) [14], the density of infected nodes varied and increased under different vaccination rates, such as λ=0.3, ε=0.21, γ = 0.1, δ=0.05 and Λ = μ=0.07. In Markov Random Field-Complex Communication Networks (MRF- CCN) [8],



((a)) Simulated Propagation Speed.

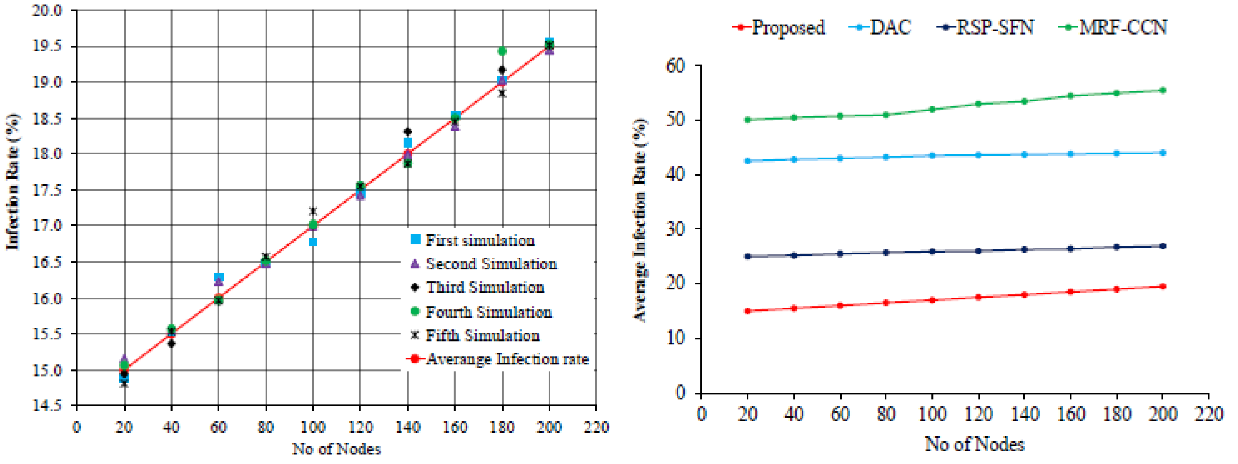
((b)) Average Simulated Propagation vs ABS-SFN [7].

Figure 8. Average infection rate.

the nodes are not reliable for long time. This leads to increasing the number of infection hosts. In our proposed scheme, the reliability is computed each time interval and also during packet transmission to monitor infection rates of the nodes in each sub-net.

4.3.3 Propagation Speed

Propagation speed was computed based on the density of nodes. The network topology greatly affects the modelling of malware propagation on IoT-based communication networks. In malware propagation, characterization of propagation speed is important. Understanding how propagation speed impacts the network is also necessary. The network size was varied in each simulation and the results of the five simulations are shown in Figure 9(a). The proposed scheme propagation speed was compared with those of the previous schemes with respect to number of nodes on varied network size as shown in Figure 10(b). In Agent-based Simulation- Scale-Free Networks (ABS-SFN) [7], the following analytical values were considered for the parameters α(k) = k−3, k = 1, 2, ...n, β = 0.3, ε = 0.01, γ = 0.08 and μ = 0.008. In addition, the reproductive ratio R0 = 3.9245 was used. If the density of infected nodes increases, the malware propagation speed also increases. The number of infected nodes increases in the ABS-SFN, whereas in our proposed scheme, the decision maker on each sub-net reduces the number of infected nodes. The proposed decision maker monitors each sub-net to determine whether it is attracted by the malware or not.



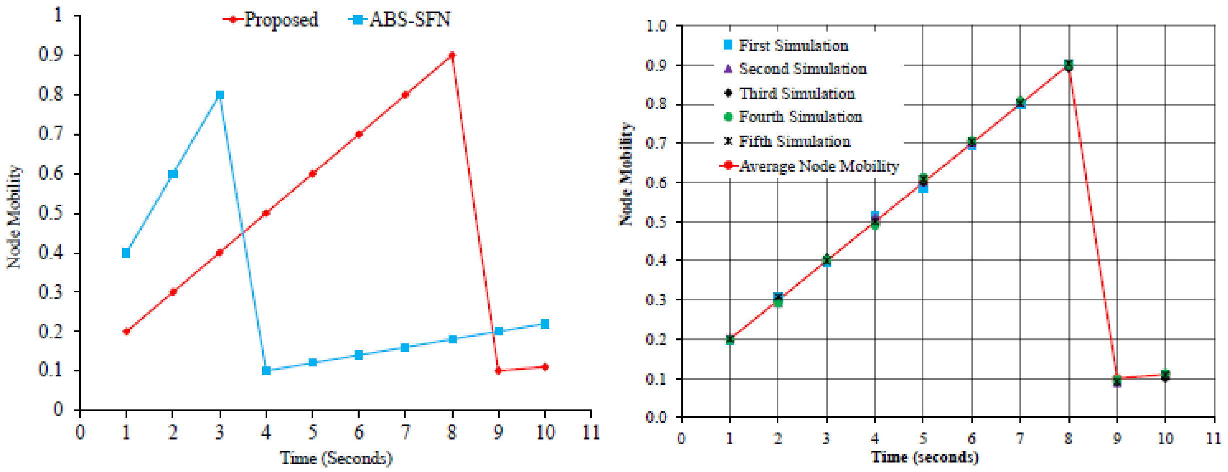
((a)) Simulated Infection Rates.

((b)) Average Infection Rate Compared with Analytical Schemes .

Figure 9. Propagation speed analysis.

4.3.4 Node Mobility

In agent-based simulation modelling, the node mobility is managed by three factors: movement detection, network connectivity or structure and location tracking. To observe node mobility, the performance at iterations i to i + 1 (between 2-4 seconds) was set in the proposed scheme. When the mobility increases above its threshold level, hub fails as noted by the decision maker and data packet transfer times between intermediate nodes are increased. In the proposed scheme, five simulations on the influence of node mobility on malware propagation were carried out. Figure 10(a) plots the mobility of nodes in a malware prone simulation against time for the five simulations. In Agent-based Simulation- Scale-Free Networks (ABS-SFN) [7], if the node mobility increases beyond the threshold, the scale-free network may disconnect. The time of the malware on the network and the malware outbreak in the sub-nets are dependent on the mobility rate. Mobility rate highly influences the spreading of network malware. When the mobility rate is smaller than the threshold value, the node in the sub-network dies. A performance comparison for node mobility between the proposed scheme and Scale-Free Networks (ABS-SFN) [7] can be seen in Figure 10.



((a)) Simulated Malware Effect on Node Mobility .   ((b)) Average Node Mobility vs ABS-SFN [7].

Figure 10. Malware effect on node mobility.

5. Conclusion and Future Work

Agent-based modelling simulation in complex networks is a challenging issue. In this paper, we developed a malware propagation model using agent-based approach and deep-reinforcement learning on a scale-free network in IoT. In the modelled system, Susceptible-Infected-Immuned-Recovered-Removed (SIIRR) transitions were formulated. The effect of malware propagation on the model was evaluated based on performance metrics, such as average energy consumption *vs.* number of nodes, average infections over time, node mobility over time period t and spreading/propagation speed. Our simulations showed that the introduction of a DRL-based decision maker results in a more versatile IoT model, where malware propagation is not just based on contact.

As future work, we intend to explore model stability analysis and the effect of immunization on different devices in IoT. The stability analysis will entail global and local model equilibrium. For the effect of immunization, we plan to incorporate mechanisms, such as targeted and proportional immunization, in the model. Employing immunization and quarantine mechanisms can offer a promising approach to make the model more realistic and resilient.

Acknowledgements

This research was partly supported by the European Commission, which is the sponsor of the METEGA project through its Education, Audio-visual and Culture Executive Agency (EACEA).

REFERENCES

[1] S.-M. Cheng, W. C. Ao, P.-Y. Chen and K.-C. Chen, "On Modeling Malware Propagation in Generalized Social Networks," IEEE Communications Letters, vol. 15, no. 1, pp. 25–27, 2011, [Online], Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5638768.

[2] S. Sneha, L. Malathi and R. Saranya, "A Survey on Malware Propagation Analysis and Prevention Model," International Journal of Advancements in Technology, vol. 6, no. 1, pp. 1–4, 2015, [Online], Available: http://dx.doi.org/10.4172/0976-4860.1000148.

[3] M. Yasir, M. A. Habib, M. Shahid and M. Ahmad, "Agent-based Modeling and Simulation of Virus on a Scale-free Network," Proceedings of the International Conference on Future Networks and Distributed Systems (ICFNDS ’17), New York, NY, USA: ACM, pp. 59:1–59:6, 2017, [Online], Available: http://doi.acm.org/10.1145/3102304.3109819.

[4] H. Kırer and Y. A. Çırpıcı, "A Survey of Agent-based Approach of Complex Networks," Ekonomik Yaklasim, vol. 27, no. 98, pp. 1–28, 2016, [Online], Available: https://www.ejmanager.com/mnstemps/94/94-1404633261.pdf?t=1552958497

[5] A. M. del Rey, A. H. Encinas, J. M. Vaquero, A. Q. Dios and G. R. Sánchez, "A Cellular Automata Model for Mobile Worm Propagation," International Work-Conference on the Interplay between Natural and Artificial Computation, Springer International Publishing, pp. 107–116, 2015, [Online], Available: http://dx.doi.org/10.1007/978-3-319-18833-1_12.

[6] L. Feng, X. Liao, Q. Han and H. Li, "Dynamical Analysis and Control Strategies on Malware Propagation Model," Applied Mathematical Modelling, vol. 37, no. 16-17, pp. 8225–8236, 2013, [Online], Available: https://doi.org/10.1016/j.apm.2013.03.051.

[7] S. Hosseini, M. Abdollahi Azgomi and A. Rahmani Torkaman, "Agent-based Simulation of the Dynamics of Malware Propagation in Scale-free Networks," Simulation, vol. 92, no. 7, pp. 709–722, 2016, [Online], Available: https://doi.org/10.1177/0037549716656060.

[8] V. Karyotis, "A Markov Random Field Framework for Modeling Malware Propagation in Complex Communications Networks," IEEE Transactions on Dependable and Secure Computing, 2017, [Online], Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7926392.

[9] K. Batool and M. A. Niazi, "Modeling the Internet of Things: A Hybrid Modeling Approach Using Complex Networks and Agent-based Models," Complex Adaptive Systems Modeling, vol. 5, no. 1, p. 4, 2017, [Online], Available: https://doi.org/10.1186/s40294-017-0043-1.

[10] A. Bose and K. G. Shin, "Agent-based Modeling of Malware Dynamics in Heterogeneous Environments," Security and Communication Networks, vol. 6, no. 12, pp. 1576–1589, 2013, [Online], Available: http://doi.acm.org/10.1145/1378600.1378626.

[11] V. Karyotis and S. Papavassiliou, "Macroscopic Malware Propagation Dynamics for Complex Networks with Churn," IEEE Communications Letters, vol. 19, no. 4, pp. 577–580, 2015, [Online]. Available: https://ieeexplore.ieee.org/iel7/4234/5534602/07029645.pdf.

[12] A. M. del Rey, A. H. Encinas, J. M. Vaquero, A. Q. Dios and G. R. Sánchez, "A Method for Malware Propagation in Industrial Critical Infrastructures," Integrated Computer-aided Engineering, vol. 23, no. 3, pp. 255–268, 2016, [Online], Available: http://dx.doi.org/10.3233/ICA-160518.

[13] E. M. Karanja, S. Masupe and J. Mandu, "Internet of Things Malware: A Survey," International Journal of Computer Science & Engineering Survey, vol. 8, no. 3, pp. 1–20, Jun. 2017, [Online], Available: http://aircconline.com/ijcses/V8N3/8317ijcses01.pdf.

[14] S. Hosseini and M. A. Azgomi, "A Model for Malware Propagation in Scale-free Networks-based on Rumor Spreading Process," Computer Networks, vol. 108, pp. 97–107, 2016, [Online], Available: https://doi.org/10.1016/j.comnet.2016.08.010.

ملخص البحث:

يُعدّ تهديد الاختراقات الضّارة عائقاً رئيسياً أمام تبادل المعلومات بشكلٍ فعّالٍ في إنترنت الأشياء. ويعدّ موضوع نمذجة الاختراقات الضّارة أحد أهم التطبيقات الملحّة الهادفة الى فهم آليات حماية بيئة إنترنت الأشياء. ويمكن تحقيق الحماية المطلوبة لإنترنت الأشياء باستخدام نمذجة قائمة على الاستفادة من وسائل الحماية في الشبكات المعقّدة.

تُقدم هذه الورقة تفصيلاتً متعمقة حول نموذج مقترح للحماية من الاختراقات الضّارة يقوم على استخدام وسائل حماية‘ إضافة الى تقنية التعلم المستند الى التعزيز العميق، في شبكة غير محددة الحجم من شبكات إنترنت الأشياء.

ويُسمى النموذج المقترح في هذا البحث طبقاً لحالات الانتقال التي يتضمنها: (مشكوكٌ فيه؛ مصابٌ بالعدوى؛ محصَّن؛ متماثلٌ "للشّفاء:؛ منزوعٌ)، وهي الحالات التي تعبر عن حالات العُقد في الشبكات المعقدة كبيرة الحجم. ويتم استقصاء موثوقية كل عُقدة باستخدام متوسط الوقت حتى الفشل. أمّا العوامل التي تؤخذ بعين الاعتبار في حساب متوسط الوقت حتى الفشل فهي: درجة العقدة، ومعدل حركية العُقدة، ومعدل الإرسال بالنسبة للعُقدة، والمسافة بين غقدتين محسوبةً وفق المسافة الإقليدية.

ويتضح من النتائج أن النموذج المقترح في هذه الدراسة قابلٌ للمقارنة مع نماذج سابقة مماثلة تتعلق بتأثيرات انتشار الاختراقات الضّارة من حيث معدل استهلاك الطاقة، ومعدل العَدْوى في زمنٍ معين، وحركية العُقد، وسرعة الانتشار.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 01, March 2020.

A REVIEW ON THE SIGNIFICANCE OF MACHINE LEARNING FOR DATA ANALYSIS IN BIG DATA

Vishnu Vandana Kolisetty and Dharmendra Singh Rajput*

(Received: 2-Aug.-2019, Revised: 26-Oct.-2019, Accepted: 16-Nov.-2019)

ABSTRACT

Big data revolution is changing the lifestyle in terms of working and thinking environments through facilitating improvement in vision finding and decision-making. But, big data science's technical dilemma is that there is no knowledge that can administer and analyze large amounts of actively increasing data and pull out valuable information. As data around the world grows rapidly and its distribution with real-time processing continues, traditional tools for automated machine learning have become inadequate. However, conventional machine learning (ML) approaches have been extended to meet the needs of other applications, but with increased information or large data knowledge bases, there are significant challenges for ML algorithms for big data analysis. This paper aims to facilitate understanding the importance of ML in the analysis of large data. It contributes to understanding the implications and challenges in big data computational complexity, classification imperfection and data heterogeneity. It discusses the capability to mine value from large-scale data for decision-making and predictive analysis through data transformation and knowledge extraction. It will suggest the impact of big data on real-time data analysis and discuss the extent to which machine learning can be used to analyze large data through machine learning in big data analysis. It will also suggest the meaning and opportunity from the point of view of encouraging feature research development in the field of ML using big data.

KEYWORDS

Big data, Machine learning, Data analysis, Big data implications, Big data challenges.

1. INTRODUCTION

In today's information world, the volume of data is bursting at an extraordinary velocity with advances in "web technologies," "social media," "mobile devices" and "sensors". Due to the multiplicity of Big Data (BD), we had to rethink the implementation of automated learning algorithms in addition to data processing framework. Choosing the right tool for an individual working situation is mostly difficult, because different types of solutions may be needed while increasing the complexity of the data itself, along with that the requirements of an automated project learning may be different.

BD has tremendous potential for commercial significance in a diversity of areas, such as "healthcare," "transportation," "e-business," "power supervision" and "economic services" [1]-[3]. But, when faced with this huge amount of data, the traditional approach suffers to perform data analysis. Research performed by "ABI (Advance Business Intelligence) Research" [4] approximates that over 30 billion interconnected devices will be there for information need. These real systems can generate enormous quantity of data from numerous resources, making it complex and difficult to perform data management, processing and analysis. It is a difficult problem for several industries and organizations to incorporate today's "healthcare companies," "IT departments," "government agencies" and "research institutions". To solve such kind of problem, a separate area was created for BD science and new trends are needed for research and education efforts [5] for rapid and successful development.

BD analysis utilization and performance of Machine Learning (ML) depend on the algorithms as well as on the setting of the applied dataset that requires a lot of time-consuming operations. In fact, some systems cannot guarantee good performance without adjusting the module. BD solutions are of high performance in a short time by providing new scientific innovations that can be integrated with ML systems for decision making. In various studies, ML is believed to be an influential tool for handling BD. As presented in [3], it is similar to the relationship between BD and the ML association among the sources and individual learning. From this perspective, individuals are able to learn from the sources to deal with innovative problems. Similarly, they are able to solve new problems through learning from BD. More information on BD processing using ML can be found in [6]-[7].

V. V. Kolisetty, Research Scholar in VIT Vellore TN, India, Email: kvishnu.vandana2016@vitstudent.ac.in
*D. S. Rajput (Corresponding Author), Associate Professor, VIT Vellore TN, India, Email: dharmendrasingh@vit.ac.in

Most of the past research works described in [8]-[9] suggest that it is difficult to perform classification of BD, as it is distributed among diverse categories of data and extracting constructive knowledge from large and composite datasets is not an easy task. BD classification demands a technique that is able to manage setbacks reasoned because of the BD attributes of "volume," "velocity" and "variety" [5]. It also needs a few calculation models and procedures to efficiently categorize data utilizing suitable ML algorithms, as discussed in various proposals [5]-[9].

Current technology development includes the latest distributed file systems and ML approaches. One such technique is "Hadoop" [10], which facilitates ML deployment utilizing exterior libraries, such as the "scikit learning library", to handle BD. Many of the ML techniques in the library mostly rely on classification algorithms which might not be appropriate for BD processing. Nevertheless, several techniques, such as "decision tree learning" and "deep learning," are appropriate for BD classification and can help develop better-supervised learning skills over the coming development periods.

The rest of the paper is organized in the following sections. Section 2 discusses big data implications. Section 3 presents data transformation and knowledge extraction. Section 4 discusses machine learning in big data analysis. Section 5 shows the importance of ML's advantage in big data. Finally, Section 6 presents the conclusion of the paper.

2. BIG DATA IMPLICATIONS

The concept of BD is initially defined as high "volume," "velocity" and "variety," but later "veracity" [11] and "value" [12]-[13] have been added. The definition needs novel processing models to facilitate visibility detection, advanced decision-making and data processing. However, "value" is characterized as the needed results to handle BD [14] and not as one of the specified BD properties. The potential of BD is highlighted by definition; however, its achievement depends on the improvement of traditional approaches or the development of new methods capable of handling this data.

2.1 Challenges

The method of supervising and utilizing a large volume of data for proposing algorithms for active and proficient methods of large data can create distinctive challenges. The challenges and modern techniques currently included in BD analysis were reviewed by Chen and Zhang [15]. Jin et al. [16] addressed the importance and opportunities of the BD concept. They also presented the challenges encountered in terms of data, order and computational complexity and suggested possible solutions to these challenges.

2.1.1 Computational Complexity

One of the major challenges faced in BD computation complexity is due to a straightforward increase in data volume. As a result, when it develops into a large size, the utilization of trivial systems is expensive and even the current ML algorithms also show a significant time complexity based on various data sample features. In case of utilizing ML algorithms like "support vector machine (SVM)", complexity is faced during the training phase of "$O(m^3)$" time and "$O(m^2)$" in the space of complexity [17], where m is the iterations needed for the training samples. Thus, the impact of m will significantly influence the time and memory requirement for training BD, rendering the process impractical.

Causes of challenges are mostly classified as: "classification," "scalability" and "analysis" based on the task to perform. In terms of technological challenges, these are classified as: "computation," "communication" and "storage". Also, with increased data size, the performance of algorithms becomes additionally reliant on the structure used to store and transfer data. As a result, the data size does not only affect performance, but it also leads to the need to revise the general architecture used to implement and develop these algorithms. Thus, with all these algorithms, as the data size increases, the time required to perform the calculations can increase dramatically and the algorithm can become unusable for very large datasets.

2.1.2 Classification Imperfection

The classification process implements methods to collect input data, understand data, transform data and understand the BD environment based on hardware necessities and acceptance criteria. Ultimately, the success of BD classification requires an understanding of modeling and algorithms. However, certain parameters affecting the classification of BD cause problems in the development of learning and classification imperfection models.

Classification imperfections are not limited to BD and have been the subject of research for more than 10 years [18]. According to experiments performed by Japkowicz and Stephen [18], the difficulty of the problem of imbalance depends on the complexity of the task, the degree of inconsistency in the classes and the total data size of the training. They recommended that the class is likely to be represented by a reasonable number of samples in a large dataset. However, an evaluation of the actual BD set is required to confirm these observations. In such a case, the complexity of BD operations is expected to increase, which can have serious consequences due to class discrepancies.

The larger the dataset, the more often it is broken, assuming that the data is evenly distributed among all classes [19]. This causes that the classification is incomplete. The performance of the ML algorithm will adversely be influenced when the dataset contains data for a class that has a variety of possible occurrences. This problem is particularly noticeable when various classes are characterized based on several samples and few are represented as extremely small numbers. As a result, in the BD context, the probability of class imbalance is high due to the size of the data. Also, because of complexity of data, the potential impact of class imbalance on the ML approach is significant.

2.1.3 Data Heterogeneity

BD analysis involves incorporating various data from multiple sources. Such data can vary depending on the data type, format, model and meaning. In practice, most real data analysis problems are caused by heterogeneous data [1], [20], different in type, structure and distribution due to the massive quantity of data composed from various sources with no class label information. For instance, in an emotion exploration activity, the data can be included as "text," "images" and "videos" collected from different social media sources. To extract knowledge from such large and unlabeled data, advanced autonomous learning technologies must have various models which are able to perform efficient integration and learning with minimum time and process complexity.

In statistics, heterogeneity defines the differences between statistical features in different datasets. These problems exist with BD as well as in small datasets, but the datasets usually contain parts from several sources. This statistical heterogeneity splits the familiar ML hypothesis that statistical features are related in an entire dataset.

In real-time applications, learning from heterogeneous sources is associated with significant challenges due to data dimensionality, multipart relationships, several structures having various objects and diverse distribution. In most cases, label learning through supervision for heterogeneous data is not presented or is time-consuming. In this case, the guidelines for heterogeneous information integration are missing and most learning methods fail to perform accurately. So, identifying an unsupervised function that will be beneficial to the overall analysis is still an important and crucial research problem.

3. DATA TRANSFORMATION AND KNOWLEDGE EXTRACTION

ML often requires data pre-processing and cleaning steps to configure the data for a particular model. However, in the case of data from different sources, the formats of the data may be different. In the context of data analysis, "data," "information" and "knowledge" are three foremost observations to be exploited. It is possible to perceive data analysis, which is able to transform and integrate data into information and can be used for visualization or decision making, as shown in Figure 1.

In an effort to optimize BD for data extraction and transformation, it is necessary to try to modify the data to become analysable by ML. This amendment process is in the pre-processing phase of the data. It also undertakes the challenges to remove dirty and noisy data through the cleaning process. In this area, there is no significant development in respect to BD and it has been an active research focus in

various domains.

The three essential aspects of data influencing ML are: large quantity, dimension and various samples. Hence, two-perceptive data for learning with BD is handled by limiting the dimension and selecting the instance. Reducing dimensionality aims to set a high-resolution space on a smaller area of dimensions without much information loss. Dimension reduction mainly solves the problem of dimension curse and enhancement in processing.

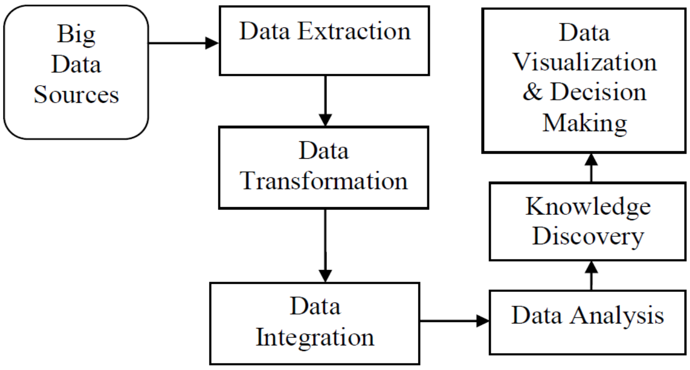


Figure 1. Big data transformation and analysis process.

The selection of instance refers to methods that select subsets, similar to the entire dataset. It is intended to reduce the dimension with large-scale datasets through data reduction and more specifically to select the required instances. The subgroup is then utilized to create conclusions regarding the entire dataset. The selection methods for various events include "random selection," "selection on genetic algorithms," "progressive sampling using domain knowledge" and "cluster sampling" [21], [46].

Data integration and management are critical issues in BD distribution. These are among the original activities utilized to advance the quality of distributed data in independent data sources. A traditional data collection system is a system that integrates the limited resources and usually has complex and time-consuming functions. As discussed in [22], data integration systems need to address uncertainties about semantic assignments between data sources and the intermediate schema in order to effectively index the keywords of the data access queries. This means that appointments are detected by understanding the meaning behind the tagging features of the elements of the schema elements, but many challenges are faced to understand the features that are reliable [23] and this is associated with BD integration [24].

3.1 Author Data Transformation

Data transformation converts data or information from one structure into another, generally from the structure of a source system to the necessitated structure of a required target data structure. Mostly, the standard procedures are engaged in converting text data files. However, during data conversion, for a while, a program is converted from one workstation execution program into another, so that the program can be executed on a diverse environment. The common motivation for this data relocation is the introduction of the latest system that is fundamentally dissimilar from the earlier systems.

In practice, data transformation engages the utilization of an exceptional program that reads the original base language of the data, determines the language in which it must be converted into a new program or the data that the system can utilize and then continues with data transformation.

Data Transformation engages two basic stages:

- ***Data mapping***: Assignment of components to capture all transformations that occur at the source base or from system to destination. This is organized for more complex systems when there are multifaceted transformations, such as multiple individuals or multiple regulations for transformation.

- ***Code generation***: Creating the original transformation program. As a result, the specification of the data map is utilized to produce carry-out programs for running on systems.

3.2 Data Analysis

Data analysis and data mining from a division of "Business Intelligence (BI)" that includes "data warehousing," "database management systems" and "online analytical processing (OLAP)". Data mining is a specific data analytic strategy that targets predictive statistical modeling and information discovery, relatively entire expressive reasons, while BI covers data analysis aiming primarily on BI.

Deeper data analysis is able to reveal many of the most important features of data, which helps predict future data features. This allows to explore the development of patterns from a set of data to a BD set. Statistical and engineering features are key analytical bases that assist us to recognize the development of patterns. One area focused on BD classification is the development of the technology sector, where the fundamental elements of analysis should be clearly understood. Some numerical evaluations contributing to these goals are: "counting," "mean," "variance," "covariance" and "correlation" [25].

The methodologies to process data for data analysis need to follow these steps:

- ***Data requirements*: **Data is required as the input of analytics, which is particularly dependent on the needs of the analytics or clients' usage. The common individual on which data accumulated is identified as the testing unit. In particular, a demographic variable (e.g., height and weight) is obtained. Data can be statistical or definite.
- ***Data collection*: **Data is accumulated from various sources corresponding to data analysts at an organization. Data can also be gathered through sensors in the surrounding, such as "traffic cameras," "satellites," "recording devices," …etc. It can also be gathered through "interviews," "downloads from web sources" or "interpreting documents".
- ***Data processing***: The data primarily acquired must be processed or organized for analysis. For example, this might include data placed in tables and columns, such as spreadsheets or statistical software in tables and columns for further analysis.
- ***Data cleaning*: **This will process and classify data which might be imperfect, duplicate or enclosing errors. Data is accessed and stored showing the need for data cleansing from problems. Cleaning data is the process of avoiding and correcting such inaccuracies. In general, it consists of "record matching," "recognizing data incompleteness," "eminence of existing data," "transcription" and "column segmentation".

Moreover, as we have already noted in the context of BD, the challenges of data classifying and cleaning are becoming more common and more difficult. Therefore, it is difficult to identify such problems and separate them to represent a complete group. In the case of large inconsistency between data rows, the process of data selection is not able to guarantee accurate class selection solutions.

4. MACHINE LEARNING IN BIG DATA ANALYSIS

ML is a division of artificial intelligence, which consists of two phases: "training" and "testing" [3]. The primary phase proposes a learning mechanism based on some of the known characteristics of datasets. The second stage aims to make predictions of unidentified characteristics through the knowledge gained in the primary phase.

In this view, "training" and "testing" are also called "learning" and "prediction". In fact, the task of ML is to use a learning algorithm to build a model that is also applied to make predictions. Therefore, this activity is generally called predictive modeling. The phases of ML from data acquisition to constructing a predictive model are shown in Figure 2.

In recent literature, several researchers illustrated ML challenges with BD [26]-[28], while others examined them in terms of a particular methodology [26]. According to [28], ML algorithms are able to develop in numerous kinds of learning, such as "Decision Tree Learning," "Rule Learning," "Instance-based Learning," "Bayesian Learning," "Perception Learning" and "Collective Learning". All these learning algorithms reflect the nature of the promotion.

In ML, there are several algorithms for constructing the model, where the word algorithm points to the learning algorithm. In this scenario, the model is treated as information modified from training data. The testing phase aims to transform information into knowledge. The learning algorithm utilizes a given set of data to learn, validate and test the model. It discovers the best value for the parameter to validate and evaluate the enhancement.

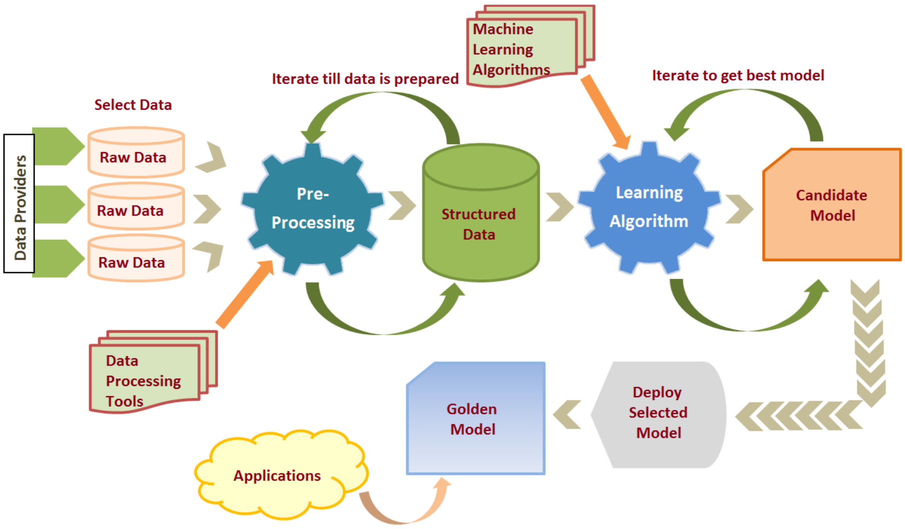


Figure 2. Machine learning phases of processing [44].

4.1 Supervised and Un-supervised Learning

Supervised Learning (SL) proposes the methods of studying with the trainer, since in all the cases, the training clusters are categorized to predict the outcomes accurately. In other words, the proposed learning is usually inspired by learners' learning under the control of supervised trainers. In doing so, the purpose of this kind of learning is to build a model by learning through accurate data and making other predictions and unrelated cases in terms of the expected attribute value. Therefore, SL can be part of the "classification" and "regression" functions for final prediction and statistical prediction, respectively [47]-[48].

In SL, classes are known and class boundaries are well defined in a given set of learning data and learning is carried out using these classes. Classification problems can be solved precisely depending on the knowledge transformations revealed above. A flowchart of supervised ML approach is shown in Figure 3.

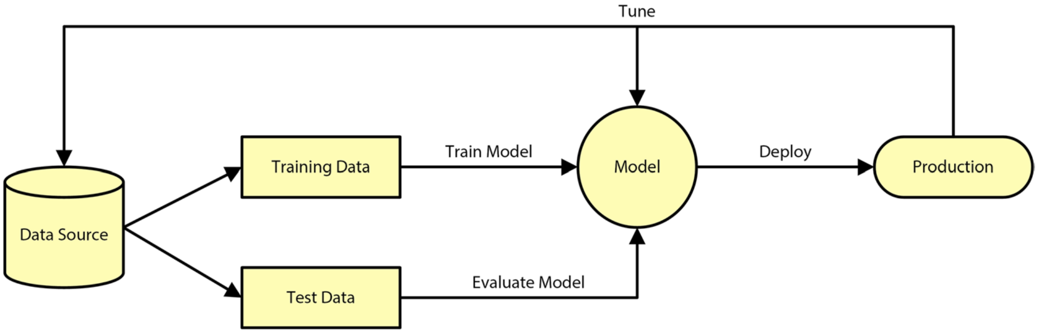


Figure 3. A flowchart of supervised ML.

Let's assume a dataset is specified and its data domain is *D* is R^c, which implies that the occurrences in the dataset are based on the ***c*** properties and create a "*c*-dimensional vector space". If it is supposed that there are "*n* classes," the function of knowledge can be given using Equation (1).

$$f: R^c \Rightarrow \{0, 1, 2, \ldots, n\} \qquad (1)$$

In Equation (1), the series from *"{0,1,2, . . . ,n}"* includes the groups of knowledge which allocate the

distinct values of labels *"0,1,2, . . . ,n"* to dissimilar classes. This mathematical purpose assists to describe the classification criteria that are appropriate for data classification. A number of classification procedures have been recommended in the ML document and a few of the well recognized methods are "SVMs" [29], [52], "decision trees" [30], "random forests" [31] and "in-depth learning" [32].

Un-Supervised Learning (USL), on the other hand, means learning without learning. This is because the learning results are not clear. In other words, learning without supervision is naturally inspired by learning. In fact, the purpose of this type of learning is to discover previously unknown dataset patterns through association and cluster insertion. The first aims to identify the relationship between the objects and attributes and the second aims to cluster the items based on their similarity.

In USL, suppose that class boundaries are unknown; so, the class labels themselves have been learned as well and classes are defined accordingly. Thus, the class boundaries are statistical and not clearly described; known as "clustering". In the clustering problem [33], it is assumed that the dataset can be created, but not categorized. As a result, it can only generate approximate rules to help categorize new data that does not contain labels. Clustering forms a guideline that facilitates labelling the selected data points and assigning labels to the new data points. As an outcome, the data can simply be collected without being classified. Therefore, clustering problems are expressed using estimation rules [49].

Clustering difficulties can also be mathematically solved based on the knowledge of data transformation, as discussed previously. Let's suppose a domain "D" with the set of data records, having c depending features, can be represented as "R^c" and forms feature vectors with a c-dimension space. To construct a cluster for the k classes, a knowledge-based function can be derived as given in Equation (2).

$$f: R^c \Rightarrow \{0, 1, 2, \ldots, k\} \tag{2}$$

The series of knowledge set is illustrated for k labels as "*{0,1,2, . . . , k},*" each label having different features. Based on these most associated features of labels, a suitable class is assigned to have accurate clusters. Few clustering algorithms in ML generally used are "*k*-Means clustering," "Gaussian mixture clustering" and "hierarchical clustering" [34].

4.2 Big Data Analysis

Business Intelligence is an application that can benefit from BD techniques. BD analyses also have systematic consequences in today's uses; hence, it is suitable to recognize them utilizing the features of the classes, the characteristics of the parameters and the characteristics of the observations; three important ideas of BD. A full understanding of the features of the classes, the characteristics of the parameters and the properties of the observations can support in addressing these problems.

Assuncao et al. [35] reviewed the development methodology and environment for performing BD analysis on the cloud platform. They categorized the BD analytics solutions "based on past customer activity -description models, "based on available data- forecast models" and "prescriptive models for supporting decision-making processes".

Personalization of acceptance and non-cooperative attempts can lead to difficulties in the BD area. Every acceptance will contribute to the BD and influence the uniqueness of the other orthogonal acceptances, thus determining acceptance problems using a three-dimensional space. This recommends that the classification of categories with BD development is very complex and unpredictable. Thus, an increase in the class forms depends on the scheme, irrespective of user knowledge and experience. Thus, BD classification becomes unpredictable and it is difficult to apply ML models and algorithms efficiently.

Similarly, the acceptance of the features contributes to BD complexity. It builds a classification utilizing the patterns to reduce complexity with growing data dimensions. These are considered as main factors that solve the scalability problem of the BD paradigm and its confirmation contributes to the complications in the data management, processing and analysis. Its expansion will increase data size and make processing difficult with current technologies in the near feature.

4.3 ML Modeling and Algorithms Approaches in Big Data

ML has different learning paradigms; however, not all these types of research are appropriate. Modeling and algorithms are defined based on the characteristics of "domain distribution," "batch learning" and "online learning" depending on the availability of data-level labeling and supervision and USL. The two foremost elements that help accomplish ML goals are aimed through learning models and learning algorithms utilizing different pattern recognition tools. Some of the tools utilized in BD for data processing are described in Table 1.

Table 1. Comparison of various BD tools.

BD Tools	Description	Advantages	Disadvantages
Apache Hadoop [62]	• It is one of the most prominent and used tools in the BD industry with a huge capacity for large-scale processing of data. • It processes large datasets through programming models, such as "MapReduce". • It is a 100% open-source framework and executes on product hardware in current data centers.	• It offers a robust ecosystem that best suits the analytical requirements of the developer. • It conveys elasticity and faster data processing. • Highly-scalable and highly-available service to rest in a cluster of computers. • The main strength of Hadoop is HDFS, which is capable of holding all types of data - video, images, JSN, XML and plain text in the same file system.	• Sometimes, disk space concerns are possible to be met due to its "3x" data redundancy. • I/O operations have to be optimized for better performance.
Apache Spark [63]	• It is the industry's next hype in BD tools. • It is an alternative to the MapReduce of Hadoop. • It is an added point for data analysts to handle definite kinds of data to accomplish quicker results.	• It is easy to execute on a particular local system to facilitate progress and testing. • It can run 100 times faster than a map of Hadoop. • It is easy to work with HDFS in addition to other data stores; for instance with "OpenStack Swift" or "Apache Cassandra". • The main point of this open-source BD tool is that it fills the gap in "Apache Hadoop" with regard to data processing. • It is capable of managing batch data and real-time data together. • It processes data quicker than conventional disk processing techniques because of in-memory data processing.	• It has no support for real-time processing. • It has no file management system and is expensive. • It has problems with small files. • Its number of algorithms is very few and it shows latency. • Manual optimization. • Iterative processing.
Apache Storm [64]	• It is a distributed real-time framework to consistently process unbound data streams. • Its topology can be considered as a MapReduce work. • It's a free and open-source BD computation system. • It can interfere with "Hadoop's HDFS" with adapters as required, which is an additional feature that builds it as an open-source BD tool.	• Its framework supports any programming language. • Depends on the topology configuration, it allocates the workload to the scheduler nodes. • It recommends distributed, real-time, fault-tolerant processing systems with real-time computation potential.	• Difficult to learn, use and debug. • The use of native scheduler and Nimbus becomes a hindrance.
Cassandra [65]	• It handles a distributed kind of database to process a big group of data on servers. • It is one of the best BD tools that mainly processes structured datasets.	• Its design architecture does not function as the master-slave architecture and every node functions as an identical role. • It is able to manage various synchronized clients across the data center.	• Troubleshooting and maintenance require some extra effort. • The process of clustering requires improvement.

	• It provides a highly available service with a single point of failure. • Additionally, it has specific capabilities that no other related database and no NoSQL database can provide. • It supports replication across multiple data centers, providing lower latency for users.	• Its database is extensively utilized at a moment to give valuable administration of a huge quantity of data. • The data is routinely duplicated to several nodes for fault tolerance. • It is mainly valuable for applications that are not able to lose data still if the complete data center is stopped. • It provides agreements' and services' support, provided by other vendors.	• The row-level locking feature is unavailable.
RapidMiner [66]	• It is a software stage for information science performance and presents a combined situation. • It follows the model of a client's server, where the server is perhaps situated on a pre-basis, or in cloud communication. • It is developed in Java and presents a GUI for designing and performing workflows. • It is able to give 99% of progressive solutions.	• It's an open-source BD tool. • It is utilized for "data preparation," "machine learning" and "model deployment". • It provides a collection of products for setting up the latest data mining procedures and projecting analytics. • It stores streaming data in numerous databases. • It allows for various data management approaches with batch processing and GUI Interface.	• Improvision in online data services is needed.
Hive [67]	• It is an open-source tool for BD. • It helps the developer perform BD analysis in Hadoop. • It assists to quickly find data search and manage large datasets.	• It maintains the SQL type query language for communication and data modeling. • It lets you define functions with Java or Python. • It is created to handle and search only structural data. • It's based on "SQL-inspired language" that sets the consumer apart from the complexities of map reduction programming. • It presents a "Java Database Connectivity (JDBC)" interface for programme integration.	• It is not designed for Online Transaction Processing (OTP). • It can be used for Online Analytical Processing (OLAP). • It doesn't support updating and deleting, but it supports overwriting or data capture. • Basically, the Hive subqueries are not supported.

Depending on the characteristics of the divisions of the field, "regression," "classification" and "clustering" might determine the modeling features of ML by supervised and unsupervised algorithms of ML [36]-[37]. Domain segmentation can also be essential in determining the learning algorithms. Suppose that the field is categorized and group labels are introduced, so that a classification model can be set up and the acquisition of optimal parameters can be monitored. It is therefore referred to as SL and classifications are defined under the SL model. If the field is separable and the class labels are not assigned, it is referred to as USL and then assigned to the USL format.

4.3.1 Supervised Learning Models

Models of SL provide parameters to move the data field for a response group, thus helping to take the knowledge from the data. These learning models are generally combined into predictive models and classification models. The "regression model" is a predictive variable that is appropriate for systems that generate continuous reactions. There are various regression models, including "standard regression," "ridge regression," "lasso regression" and "elastic-net regression" [38]. In this model, the factor creates an important function in reducing the error to the incline factor and the normalization factor.

The classification model is suitable for scenarios where individual results are created. There are many classification representations that can be grouped under "mathematically intensive," "hierarchical models" and "hierarchical models". Hierarchical models assist to classify separated group points associated with base classes utilizing a tree-like structure [50]. This model is well suited for modern requirements, including BD and distributed ML. It adopts together regression analysis and classification approach using trees that can be constructed with a series of decisions, called decision trees.

4.3.2 ML Supervised Learning Algorithms

SL algorithms assist in model training effectively to provide high-grade accuracy. In general, SL algorithms support the use of large datasets to retrieve optimal values for model parameters without over-installing the model. Therefore, it is important to carefully design the learning algorithm using a systematic approach. The ML field proposes three phases of designing an SL algorithm as, "training phase," "verification phase" and "testing phase".

Training algorithms mainly help adjust and optimize model parameters using categorized datasets. The training algorithm needs "quantitative measures" to effectively train the learning model by means of the distinctive marked dataset. In general, it includes several sub-processes, such as extracting the data field and creating the associated group, standardization and modeling. Model testing is a procedure to evaluate the enhancement of a model that has been trained with a training algorithm. Few such algorithms based on training are described below.

- ***Support Vector Machine:*** This method helps in resolving one of the BD classification issues in a classic ML technology. Specifically, it can help in multiple domain applications in the BD environment, but it is complex during computation. It is utilized in BD frameworks, like "RHadoop," based on SVM implementation with R-programming for analyzing distributed file systems. Even for "MapReduce" in Hadoop framework SVM [53], associated algorithms are deployed to improvise the functions.

- ***Decision Tree Learning:*** Decision trees use rule-based approaches to divide domains into several linear spaces and predict reactions. If the predicted reaction is repetitive, the decision tree is a "regression tree" and if the predicted reaction is individual, the tree is a "classification tree". In fact, "decision tree-based learning" management is described as a "rule-based binary tree" creation procedure, but it is easy to recognize if it is interpreted as a hierarchical field partitioning system. The data area is recursively partitioned into two sub-domains to obtain more information gain than in the partitioned node approach. Decision trees are able to be "trained," "verified" and "assessed" exploiting SL algorithms, so it is clear that they form an SL model and satisfy this definition.

- ***Random Forest Learning:*** This learning method utilizes the decision tree modeling approach [3]. This technique utilizes a decision tree model for parameterization, which includes "sampling techniques," "sub-space techniques" and "ensemble techniques" to optimize modeling, which is generally called bootstrap modelling and is substituted with a "random sampling method". Based on this, it supports to construct and choose a decision tree for random forest configuration. This decision tree can be either in the form of a "classification tree" or a "regression tree". Hence, it can be mutually useful for classification and regression issues.

- ***Deep Learning Models:*** Deep learning models in ML try to understand the relations embedded in learning representation. This is mostly expressed by the frequently used term "learning by features" [40]. This kind of algorithm takes its name from the reality that it utilizes data representation rather than precise data functions to execute jobs. It transforms data into abstract illustrations that facilitate learning. In a deep-learning structure, these presentations will later be used to perform ML tasks. Since the functions are discovered directly from the data, the parameters do not need to be configured. In the BD context, the ability to avoid technical features is an immense benefit because of the challenges correlated with this process.

Deep-learning algorithms able to confine different stages of abstraction. This type of learning is, therefore, the best clarification to the "image classification" and "recognition problem". "Boltzmann machines" [41] are related with the exception that they use a random rather than an inevitable process. Another example of these algorithms is "deep-belief networks" [42]. Because of the illustrated features, deep learning appears to be well suitable to handle several predefined challenges, such as "geometry features," "data heterogeneity," "nonlinearity" and "noisy data". However, these algorithms are not designed primarily for varying and volume data learning [43] and therefore prone to the data speed problem. While they are well-suited to handling large amounts of data with complex problems, they are not computationally efficient [45].

Najafabadi et al. [26] focused on deep learning, but pointed out the common disadvantages of ML with BD: "unstructured data formats," "fast data streaming," "multi-source data entry," "noisy" and "bad data," "high dimensions," "scalability of algorithms," "unbalanced input data" and "limited labeled data". Similarly, Sukumar [27] recognized three main prerequisites: "designing flexible" and "highly scalable structures," "understanding the properties of statistical data" before applying algorithms and ultimately developing the capacity to work with large datasets. In Najafabadi et al. [26] and Sukumar [27], investigations reconsidered ML characteristics with BD, but did not do effort to link every acknowledged challenge.

Qiu et al. [28] developed various learning methods and presented various works of BD. Although they performed an immense job on current issues to identify possible solutions to the lack of classification as well as on approaches to solve the challenges and deepen the relationship between the hard-informed decision-making model and the learning outcomes which are most suitable for a particular task or a specific scenario. Thus, the focus of our work is to establish a link between solutions and challenges. A comparative analysis of these proposals and their limitations is presented in Table 2.

Table 2. Comparison of proposal enhancements and limitations.

Author	Approach	Datasets Used	Enhancement	Limitation
L. Xiang et al. [1]	Two-stage unsupervised multiple kernel extreme learning machine	UCI machine learning repository	Flexible algorithm for fast unsupervised heterogeneous data learning	High computational overhead
H. Liu et al. [6]	Predictive modelling, Decision tree, Bayesian and Instance-based learning	UCI and biomedical repositories	Building accurate, efficient and interpretable computational models	High-variability data showing high variance in terms of accuracy performance
I. W. Tsang et al. [17]	SVM and a core vector machine (CVM) algorithm.	KDDCUP-99 intrusion detection data	Optimal solutions for efficient classification with the use of core sets	High expense because of time and space complexities
M. Ghanavati et al. [19]	Integrated method for learning large imbalanced datasets	Water pipeline datasets	Effective for the well-learned datasets.	Ineffective for big and highly imbalanced data
C. Zhu et al. [20]	Heterogeneous metric learning with hierarchical couplings	30 datasets from different domains	Solution for complex categorical data with hierarchical coupling relationships and heterogeneities	Limited to specific data characteristics and domain knowledge

H. A. Mahmoud et al. [22]	Probabilistic model based on Naive Bayes classification	2323 schemas from 5 different domains from Google's web	The performance of the clustering algorithm shows increases in precision and recall	No comparison is shown with the existing clustering algorithms
N. Ayat et al. [24]	IFD (Integration based on Functional Dependencies) with a probabilistic data model	Dataset of the university domain	Significant performance gain in terms of recall and precision compared to the baseline approaches	No measure has been shown to enhance the integration of uncertain data
J. Read et al. [43]	Deep-learning techniques	Real-world datasets	Improvement in the accuracy of popular existing data-stream methods	No clear explanation of higher-dimensional datasets in terms of feature reduction and classification of labels

4.4 Limitations of Big Data Analytics

BD brings some big hopes. However, this is not a tool with unlimited features, making the most of the analysis means underestimating the limitations of using data capabilities [54]. The following are some of the major limitations of experienced users and first-time data explorer.

- *Data Misinterpretation*: Data can reveal the user's behavior. However, it cannot also advise why users think or behave in their ways. But, misinterpretation of data is able to misguide dealers in their business attempting to capture utilizing the market progressive information. In addition, depending exclusively on data to formulate possibility may guide companies to take actions based on wrong relevance. The actuality is that identifying the predicted correlation and attempting to respond to the correct problem in support of the data is a different job from gathering and interpretation the data.
- *Security Limitations*: BD is also facing limitations due to security issues. Companies that collect data have a significant responsibility to protect data. The consequences of data breaches may include litigation, fines and loss of reputation. Security issues can greatly inhibit your ability to process data. For example, analyzing data by other organizations can be complicated, since the data might be concealed with a firewall or private cloud server. This creates a lot of trouble for sharing and transmitting data to be analyzed and worked on in a reliable manner.
- *Outlier Effect*: The third major constraint with BD is that outliers are common. Once the data is processed and analyzed, the user's failure or a new upgrade to the popular search engine will produce some biased results. The reality is that technology is not yet able to collect data completely accurately. However, Google's own algorithms and the inability to correctly predict search behavior made the project one of the company's most compelling failures to date.

5. FEATURE SIGNIFICANCE OF ML IN BIG DATA

ML utilizes an algorithm to discover hidden knowledge without explicit programming. In ML, it is important to understand repetitive components, where the models tend to adapt independently when exposed to BD. So, with the advent of new computer technologies, ML has significantly advanced from the past. Recently, ML algorithms have been able to consistently perform complex computations to integrate and analyze BD, which has not been available for a long time. A few well-known examples are illustrated below.

- The concentrate of ML with BD able to be found in "Google's self-driving car".
- ML applications utilizing BD are able to find various "recommendations" and "online business systems", such as Amazon, Netflix online, …etc.
- In-text data processing in various social media information, like Facebook, Twitter, …etc.
- ML can process BD to predict fraud detection in various financial and security systems.

The most commonly used ML methods include "SL," "USL," "class supervised learning" and "reinforcement learning". However, SL-based method utilization is nearly 70%, whereas USL utilization is about 10-20%.

- The significance of the SL algorithm is to be utilized where the required result is well-known. This algorithm is used with a set of inputs and a corresponding set of outputs. The algorithm compares the actual output with the correct output in feature analysis.
- Unsupervised learning is utilized for data without historical label. The algorithm must know that it is displayed to give the correct result and semi-SL is utilized for labeled and unmarked data, such as "classification," "regression" and "prediction".
- USL is utilized in opposition to data with no past labels. This algorithm must predict the correct result without knowing the data labels.

5.1 Future Research Directions of Big Data Analysis

Today, BD analysis is getting more and more attention, but there are still many research problems to be solved in various domains.

- *Storage and Retrieval*: Multidimensional data has to be integrated with the analysis on BD; so, you can explore arrays depending on in-memory illustration models [55]. The incorporation of multidimensional data representations on BD involves the use of multidimensional extensions to enhance the query language HiveQL. With the rapid development of smartphones, images, audios and videos are produced at an alarming rate. However, the storage, retrieval and processing of this unstructured data require extensive research in various dimensions [56].
- *BD Computations*: In addition to the current BD paradigms, such as "MapReduce" [51], other paradigms are relevant, such as "YarcData (BD Graph Analytics)" and "high-performance computing cluster (HPCC)" systems need to be investigated [57].
- *Visualization of High-Dimensional Data*: Visualization facilitates assessment analysis at each action of data analysis. It concerns the remaining fraction of the "data warehousing" and "OLAP" research. For high-dimensional data, a various range of visualization tools is being developed [58].
- *Real-time Processing Algorithms*: Due to the frequency at which data and forecasts are produced, the various real-time algorithms might not be able to realize the processing time complexity and delay.
- *Social Perspective's Dimension*: It's essential to recognize that various technologies are able to produce quicker results, but assessment makers must utilize them intelligently [59]. These outcomes may possibly have some social and cultural influences. There is no doubt that large-scale search data will assist in generating improved tools and facilities, as well as privacy intrusion and intrusive marketing. Data analysis assists even in understanding online behavior, local community and political movements [60]-[61].

6. Conclusion

BD analysis is the process of examining large and diverse datasets. Learning from large, unstructured data offers significant opportunities for many sectors. However, most of these routines are not sufficiently computational, practical or scalable. This paper presents a review of the need for research aimed at proposing new techniques that can be used to analyze BD. The concept of ML is increasingly adopted in current and future trends in BD implementation. This paper presents the challenges faced by various ML tools to provide an adaptable framework that fits the BD field of analysis. Analytical units can be combined with an ML engine to overcome data processing conditions. BD analytics and ML implementation support each other and can be powerful tools for understanding and predicting business behavior based on customer input information. With increasing use of ML concepts in research and business, the requirement of new methods to assist learning tasks has become increasingly essential in future research works to achieve significant improvements in ML approaches for BD analysis.

REFERENCES

[1] L. Xiang, G. Zhao, Q. Li, W. Hao and F. Li, "TUMK-ELM: A Fast Unsupervised Heterogeneous Data Learning Approach," IEEE Access, vol. 6, pp. 35305-35315, 2018.

[2] W. Raghupathi and V. Raghupathi, "Big Data Analytics in Healthcare: Promise and Potential," Health Information Science Systems, vol. 2, no. 1, pp. 1-10, 2014.

[3] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis and K. Taha, "Efficient Machine Learning for Big Data: A Review," Big Data Research, vol. 2, no. 3, pp. 87-93, Sep. 2015.

[4] ABI Research, "Billion Devices Will Wirelessly Connect to the Internet of Everything in 2020," [Online], Available: https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne/, 2013.

[5] P. Zikopoulos and C. Eaton, Understanding Big Data: Analytics for Enterprise-class Hadoop and Streaming Data, McGraw-Hill Osborne Media, 2011.

[6] H. Liu, A. Gegov and M. Cocea, "Unified Framework for Control of Machine Learning Tasks Towards Effective and Efficient Processing of Big Data," Springer Data Science and Big Data: An Environment of Computational Intelligence, pp. 123–140, 2017.

[7] X. Wu, X. Zhu, G.-Q. Wu and W. Ding, "Data Mining with Big Data," IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 1, pp. 97–107, 2014.

[8] S. Suthaharan. "Big Data Classification: Problems and Challenges in Network Intrusion Prediction with Machine Learning," ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 4, pp. 70–73, 2014.

[9] H. Tong, "Data Classification: Algorithms and Applications," Taylor and Francis Group, pp. 275–286, 2015.

[10] K. Shvachko, H. Kuang, S. Radia and R. Chansler. "The Hadoop Distributed File System," Proc. of the 26th IEEE Symposium on Mass Storage Systems and Technologies, pp. 1–10, 2010.

[11] R. Narasimhan and T. Bhuvaneshwari, "Big Data - A Brief Study," International Journal of Science Eng. Research, vol. 5, no. 9, pp. 350-353, 2014.

[12] W. Fan and A. Bifet, "Mining Big Data: Current Status and Forecast to the Future," SIGKDD Explorations Newslett., vol. 14, no. 2, pp. 1-5, Dec. 2012.

[13] Y. Demchenko, P. Grosso, C. De Laat and P. Membrey, "Addressing Big Data Issues in Scientific Data Infrastructure," Proc. of the International Conference on Collaboration of Technol. Systems (CTSs), pp. 48-55, 2013.

[14] M. Ali-ud-din Khan, M. F. Uddin, N. Gupta and N. Gupta, "Seven V's of Big Data Understanding: Big Data to Extract Value," Proc. Zone Conference Amer. Soc. Eng. Education, pp. 1-5, 2014.

[15] C. L. Philip Chen and C.-Y. Zhang, "Data-intensive Applications, Challenges, Techniques and Technologies: A Survey on Big Data," Information Science, pp. 314-347, 2014.

[16] X. Jin, B.W. Wah, X. Cheng and Y. Wang, "Significance and Challenges of Big Data Research," Big Data Research, vol. 2, pp. 59-64, 2015.

[17] I. W. Tsang, J. T. Kwok and P.-M. Cheung, "Core Vector Machines: Fast SVM Training on Very Large Data Sets," Journal Machine Learning Research, vol. 6, pp. 363-392, 2005.

[18] N. Japkowicz and S. Stephen, "The Class Imbalance Problem: A Systematic Study," Intell. Data Analysis, vol. 6, no. 5, pp. 429-449, 2002.

[19] M. Ghanavati, R. K.Wong, F. Chen, Y.Wang and C.-S. Perng, "An Effective Integrated Method for Learning Big Imbalanced Data," Proc. of IEEE International Congr. on Big Data, pp. 691-698, 2014.

[20] C. Zhu, L. Cao, Q. Liu, J. Yin and V. Kumar, "Heterogeneous Metric Learning of Categorical Data with Hierarchical Couplings," IEEE Transaction Knowl. Data Eng., vol. 30, no. 7, pp. 1254-1267, Jul. 2018.

[21] H. Liu and H. Motoda, "Instance Selection and Construction for Data Mining," Springer, New York, vol. 608, 2013.

[22] H. A. Mahmoud and A. Aboulnaga, "Schema Clustering and Retrieval for Multi-domain Pay-as-you-go Data Integration Systems," Proc. of ACM SIGMOD International Conference on Management of Data, pp. 411-422, 2010.

[23] A. Kadadi, R. Agrawal, C. Nyamful and R. Atiq, "Challenges of Data Integration and Interoperability in Big Data," Proc. of IEEE International Conference on Big Data, pp. 38-40, USA, 2014.

[24] N. Ayat, H. Afsarmanesh, R. Akbarinia and P. Valduriez, "Uncertain Data Integration Using Functional Dependencies," Amsterdam: Informatics Institute, University of Amsterdam, 2012.

[25] D. A. Berry and B.W. Lindgren, Statistics: Theory and Methods, 2nd Edition, International Thomson Publishing Company, 1996.

[26] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R.Wald and E. Muharemagic, "Deep Learning Applications and Challenges in Big Data Analytics," Journal of Big Data, vol. 2, no. 1, pp. 1-21, 2015.

[27] S. R. Sukumar, "Machine Learning in the Big Data Era: Are We There Yet?," Proc. of the 20th ACM SIGKDD Conference on Knowl. Discovery and Data Mining, Workshop Data Science, pp. 1-5, 2014.

[28] J. Qiu, Q.Wu, G. Ding, Y. Xu and S. Feng, "A Survey of Machine Learning for Big Data Processing," EURASIP Journal Adv. Signal Process., vol. 67, pp. 1-16, 2016.

[29] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt and B. Scholkopf. "Support Vector Machines," IEEE Intelligent Systems and Their Applications, vol. 13, no. 4, pp. 18–28, 1998.

[30] S. K. Murthy, "Automatic Construction of Decision Trees from Data: A Multi-disciplinary Survey," Data Mining and Knowledge Discovery, Kluwer Academic Publishers, vol. 2, no. 4, pp. 345–389, 1998.

[31] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun and R. Fergus. "Regularization of Neural Networks Using Drop-connect," Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 1058–1066, 2013.

[32] X. -W. Chen and X. Lin, "Big Data Deep Learning: Challenges and Perspectives," IEEE Access, vol. 2, pp. 514-525, 2014.

[33] A. K. Jain, "Data Clustering: 50 Years Beyond K-means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651–666, 2010.

[34] T.-H. T. Nguyen and V.-N. Huynh, "A k-Means-like Algorithm for Clustering Categorical Data Using an Information Theoretic-based Dissimilarity Measure," Proceedings of the 9th International Symposium on Foundations of Information and Knowledge Systems (FoIKS), vol. 9616, pp. 115-130, 2016.

[35] M. D. Assuncao, R. N. Calheiros, S. Bianchi, M. A. S. Netto and R. Buyya, "Big Data Computing and Clouds: Trends and Future Directions," Journal of Parallel Distributed Computing, vol. 79, pp. 3-15, 2015.

[36] S. B. Kotsiantis. "Supervised Machine Learning: A Review of Classification Techniques," Informatica, vol. 31, pp. 249–268, 2007.

[37] O. Okun and G. Valentini, "Supervised and Unsupervised Ensemble Methods and Their Applications," Studies in Computational Intelligence Series, vol. 126, 2008.

[38] H. Zou and T. Hastie. "Regularization and Variable Selection *via* the Elastic Net," Journal of the Royal Society Series, vol. 67, no. 2, pp. 301–320, 2005.

[39] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[40] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," IEEE Transaction Pattern Analysis Mach. Intell., vol. 35, no. 8, pp. 1798-1828, 2013.

[41] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann Machines," Proc. of the International Conference Artif. Intell. Statist., pp. 448-455, 2009.

[42] G. Hinton, "Deep Belief Nets," Encyclopaedia of Machine Learning, pp. 267-269, 2010.

[43] J. Read, F. Perez-Cruz and A. Bifet, "Deep Learning in Partially-labeled Data Streams," Proc. of the 30th Annu. ACM Symp. Appl. Computer, pp. 954-959, 2015.

[44] IMARTICUS, "What Is Machine Learning and Does It Matter?," [Online], Available: "https://imarticus.org/what-is-machine-learning-and-does-it-matter/".

[45] S. M. Basha and D. S. Rajput, "A Roadmap towards Implementing Parallel Aspect Level Sentiment Analysis" Multimedia Tools and Applications, Springer, vol 78, no. 1, pp 1-30, Jan. 2019.

[46] D. S. Rajput, R. S. Thakur and G. S. Thakur, "A Computational Model for Knowledge Extraction in Uncertain Textual Data Using Karnaugh Map Technique," International Journal of Computing Science and Mathematics, InderScience, vol. 7, no. 2, pp. 166-176, 2016.

[47] S. M. Basha and D. S.Rajput, "A Supervised Aspect Level Sentiment Model to Predict Overall Sentiment on Twitter Documents," International Journal of Metadata, Semantics and Ontologies, InderScience, vol. 13, no. 1, pp. 33-41, 2018.

[48] S. M. Basha, D. S. Rajput and V. Vandhan, "Impact of Gradient Ascent and Boosting Algorithm in Classification," International Journal of Intelligent Engineering and Systems, vol. 11, no. 1, pp. 41-49, 2018.

[49] D. S. Rajput, "Review on Recent Developments in Frequent Item Set Based Document Clustering, Its Research Trends and Applications," International Journal of Data Analysis Techniques and Strategies, InderScience, vol. 11, no. 2, pp. 176-195, 2019.

[50] S. M. Basha and D. S. Rajput "Parsing Based Sarcasm Detection from Literal Language in Tweets," Recent Patents on Computer Science, vol. 11, no. 1, pp. 62-69, 2018.

[51] F. Ö. Catak and M. E. Balaban, "A Map Reduce Based Distributed SVM Algorithm for Binary Classification," Turkish Journal of Electrical Engineering & Computer Sciences, vol. 24, pp. 863-873, 2016.

[52] L. Demidova, E. Nikulchev and Yu. Sokolova, "The SVM Classifier Based on the Modified Particle Swarm Optimization," International Journal of Advanced Computer Science and Applications, vol. 7, no. 2, pp. 16-24, 2016.

[53] J. Tian, H. Rong and T. Zhao, "Hybrid Safety Analysis Method Based on SVM and RST: An Application to Carrier Landing of Aircraft," School of Reliability and Systems Engineering, vol. 80, pp. 56-65, Dec. 2015.

[54] L. Wang, G. Wang and C. A. Alexander, "Natural Language Processing Systems and Big Data Analytics," International Journal of Computational Systems Engineering, vol. 2, no. 2, pp. 76–84, 2015.

[55] A. Cuzzocrea, I.-Y. Song and K. C. Davis, "Analytics over Large-scale Multidimensional Data: The Big Data Revolution", Proceedings of the ACM 14th International workshop on Data Warehousing and OLAP, pp. 101-104, 2011.

[56] V. Agneeswaran, "Big-data - Theoretical, Engineering and Analytics Perspective," Big Data Analytics, Springer, vol. 7678, pp. 8-15, 2012.

[57] M. Chen, S. Mao and Y. Liu, "Big Data: A Survey," Mobile Networks and Applications, Springer, vol. 19, no. 2, pp. 171-209, 2014.

[58] H. Li and X. Lu, "Challenges and Trends of Big Data Analytics", Proc. of the 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 566-567, 2014.

[59] N. Khan, I. Yaqoob, I. A. T. Hashem et al., "Big Data: Survey, Technologies, Opportunities and Challenges," The Scientific World Journal, vol. 2014, Article ID 712826, pp. 1-18, 2014.

[60] D. Jothimani, A. K. Bhadani and R. Shankar, "Towards Understanding the Cynicism of Social Networking Sites: An Operations Management Perspective," Procedia - Social and Behavioural Sciences, vol. 189, pp. 117–132, 2015.

[61] M. Blount, M. Ebling, J. Eklund, A. James, C. McGregor, N. Percival, K. Smith and D. Sow, "Real-time Analysis for Intensive Care: Development and Deployment of the Artemis Analytic System," IEEE Engineering in Medicine and Biology Magazine, vol. 29, no. 2, pp. 110-118, 2010.

[62] Apache Hadoop, "Hadoop Releases, Apache Software Foundation," [Online], Available: https://hadoop.apache.org/.

[63] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker and I. Stoica, "Spark: Cluster Computing with Working Sets," USENIX Workshop on Hot Topics in Cloud Computing (HotCloud).

[64] Apache Storm, "Apache Storm, Apache Software Foundation," [Online], Available: http://storm.apache.org/.

[65] Apache Cassandra, "Apache Cassandra, The Apache Software Foundation," [Online], Available: http://cassandra.apache.org.

[66] M. Hofmann and R. Klinkenberg, RapidMiner: Data Mining Use Cases and Business Analytics Applications, CRC Press, Taylor and Francis Group, A Chapman & Hall Book, 2013.

[67] Apache Hive, "Apache Hive, Apache Software Foundation," [Online], Available: http://hadoop.apache.org/hive.

ملخص البحث:

لقد أحدثت ثورة البيانات الضخمة تغييرات في نمط الحياة من حيث بيئات العمل والتفكير، من خلال تسهيل التحسينات في إيجاد الرؤى واتخاذ القرارات. غير أن المعضلة الفنية لعلم البيانات الضخمة تتمثل في عدم وجود المعرفة الكافية لإدارة وتحليل كميات هائلة من البيانات المتزايدة لاستخلاص معلومات قيّمة. وفي ظل النمو السريع للبيانات حول العالم واستمرار توزيعها باستخدام المعالجة في الزمن الحقيقي، فإن الأدوات التقليدية المتعلقة بتعلم الآلة لم تعد كافية. ومع ذلك، فإن الطرق التقليدية لتعلم الآلة قد جرى توسيعها لتلبية احتياجات تطبيقات أخرى، ولكن مع ازدياد المعلومات او قواعد المعرفة الخاصة بالبيانات الضخمة، فثمة تحديات تواجه خوارزميات تعلم الآلة بالنسبة لتحليل البيانات الضخمة.

تحاول هذه الدراسة تسهيل فهم أهمية تعلم الآلة في تحليل البيانات الضخمة. كما تسهم في فهم ما تتضمنه حسابات البيانات الضخمة من تعقيدات وبيان أبرز التحديات المتعلقة بتحليل تلك البيانات، الى جانب ما يرتبط بذلك من نقصٍ في دقة التصنيف وفي تجانس البيانات. وهي تناقش إمكانية استخلاص لامعنى من البيانات هائلة الحجم من أجل اتخاذ القرارات ومن أجل التحليل التنبؤي عبر تحويل البيانات واستخلاص المعرفة. وتبحث هذه الدراسة في أثر البيانات الضخمة في تحليل البيانات في الزمن الحقيقي ومدى إمكانية استخدام تعلم الآلة في تحليل البيانات الضخمة. ويؤمل أيضاً أن تكون هذه الدراسة منصّة انطلاق لدراسات وبحوث مستقبلية في مجال استخدام تعلم الآلة في تحليل البيانات الضخمة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 01, March 2020.

TOWARDS MODELING HUMAN BODY RESPONSIVENESS TO GLUCOSE INTAKE AND INSULIN INJECTION BASED ON ARTIFICIAL NEURAL NETWORKS

Amin Alqudah[1]*, Abdel-Rahman Bani Younes[1] and Ali Mohammad Alqudah[2]

(Received: 26-Aug.-2019, Revised: 4-Nov.-2019 and 25-Nov.-2019, Accepted: 30-Nov.-2019)

ABSTRACT

Diabetes is one of the most widespread diseases around the world, especially in the western world where non-healthy and fast foods are widely used. Many types of research have been conducted for developing methods for predicting, diagnosing and treating diabetes. One of the tools used for this purpose is mathematical modelling, which is used for developing models of blood glucose and insulin intake. In this paper, a model to determine the proper insulin dose for diabetic inpatients was implemented using Artificial Neural Network (ANN). The model is developed by taking into consideration ten different parameters (Patient's Gender, Patient's Age, Body Mass Index for Patient, Disease History, Total Daily Insulin Doses, Diabetes Type, Smoking Factor, Genetic Factor, Creatinine Clearance and Accumulative Glucose), in addition to real-time blood glucose readings. The model is developed based on a dataset from 159 inpatients from three different hospitals. It was found that the model with the best performance was based on one hidden layer with six neurons and seven inputs. The significant inputs were glucose readouts, glucose difference, normal range, accumulative glucose, history of the disease, total insulin dose and the patient's gender. The MSE of the best model was 5.413 and the correlation was 0.9315 with negligible training time.

KEYWORDS

Neural networks, Insulin dose, Glucose level, Modelling.

1. INTRODUCTION

In the recent decades, improvement in engineering techniques and their applications in different fields in the daily life has been noticed. Today, we can see their applications almost everywhere. The medical field is one of the widest and most important fields in engineering applications. We can see devices or equipment developed using different technologies in every hospital room, so that any medicine doctor can't do his work without using these devices [1]. During recent decades, humanity developed many devices which are used to help doctors in diagnosis and treatment, as well as in overcoming illness, body's organ insufficiency, diseases, accidents and congenital malformations. In ancient times, these devices were simple and primitive. However, mankind instinct has made it vital to be discovered [2].

One of the common diseases in current decades is diabetes, which is mainly a result of the modern life style. Diabetes has two main types; type one and type two. It infects all ages and both genders [1]-[2]. Therefore, researchers focused on using the engineering science and its applications or technologies to contribute to diabetes diagnosis and therapy. The therapy of diabetic disease involves life style change, weight loss and oral medications; but mostly it depends on insulin injection based on the readings of blood glucose monitoring devices which determine the amount of glucose in the patient’s blood [2]-[4].

Diabetes mellitus is one of the most popular diseases around the world with around three hundred forty seven million people worldwide having this illness [1]. It occurs when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin produced, where this will cause what is called hyperglycemia. Hyperglycemia can be interpreted as an increment in the blood glucose level above the normal rate. There are two main types of diabetes: Type 1 and Type 2 diabetes. Type 1 diabetes usually appears in childhood age and the patients require a lifetime insulin injection. Type 2 diabetes usually develops in adulthood and mid-age. This is the most common type, representing over 90% of

1. *A. Alqudah (Corresponding Author) and A. Bani Younes are with Department of Computer Engineering, Yarmouk University, Irbid, Jordan. Email: amin.alqudah@yu.edu.jo
2. A. M. Alqudah is with Department of Biomedical Systems and Informatics Engineering, Yarmouk University, Irbid, Jordan. Email: ali_qudah@hotmail.com.

diabetes cases worldwide [2]-[4]. The treatment of this type may involve life style change, weight loss, oral medications or insulin injection [2]. Diabetes represents a major challenge for human health in the 21st century; so, there are many studies trying to provide medical solutions to this disease [3]. For patients who depend on insulin injection, the proper insulin dose is a very important issue, where determining this dose is a diabetes consultant matter. Such consultant is not always available; so, the patient must determine the dose by himself/herself, depending on his/her experience with the behavior of his/her body, which is medically unsafe.

This problem becomes more complicated by time, because diabetics mostly suffer from a slow damage of their sensitive organs, like vision problems and decline in sight intensity; so, the ability to see the injection shot's gauge becomes difficult and patients need external help for this task which is not always available [4]-[5]. Figure 1 shows a block diagram for the blood glucose track in human body by insulin interactive role to exchange glucose to energy. The digestive track breaks down the carbohydrate in the food into glucose and glucose is stored in the liver as glycogen. If the blood glucose drops under a certain threshold, the liver releases stored glucose. In order to extract glucose from the body, the liver needs insulin, which suppresses the inverse process. Most cells need insulin to consume the necessary glucose, like muscles which produce energy. If the glucose level increases in blood above the renal threshold, the body gets rid of this glucose by urine [5].

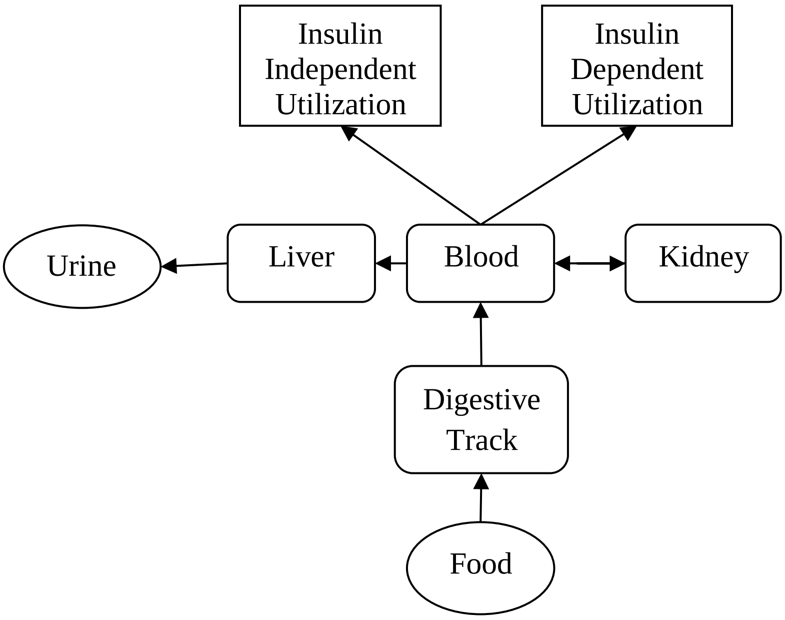

Insulin Independent Utilization
Insulin Dependent Utilization
Urine
Liver
Blood
Kidney
Digestive Track
Food

Figure 1. Glucose track in the human body, which is promoted and inhibited by insulin in the blood [5].

As research brought new renaissance in the world of manufacturing, the so-called information revolution resulted in new machines and devices. These machines and devices have artificial intelligence systems that allow to receive data and make decisions. As a result, smart devices have invaded various fields including the field of medicine and health care, providing higher accuracy in testing, measuring, supervising, controlling, organizing, alarming and achieving higher efficiency in the treatment of problems. One of the most important things that smart devices improved was the time needed to perform tasks; some testing that needed a day or even more can now be done in a few seconds. For example, blood analysis, which causes a lot of pain and suffering, is now carried out saving effort and cost and above all saving lives. Currently, there are many automatic medical devices used to monitor or treat diabetes disease. The most famous of these devices are:

1.1 Blood Glucose Monitoring Devices (BGMDs)

Blood glucose monitoring devices are considered very important for diabetic patients to monitor and manage their cases. These devices are widely used and easy to use, with both high accuracy and low cost [6]-[7]. BGM strips are used to withdraw the blood sample which is tested to measure the blood sugar ratio.

1.2 Insulin Pumps

An insulin pump is a small device which is controlled by a programmed microcontroller to deliver insulin doses to diabetic patients. These doses are delivered by a catheter, which is a flexible plastic tube. This catheter is inserted through the skin and placed into fatty tissues [8].

There are many advantages of using insulin pumps, like reducing unnecessary insulin doses. Insulin is delivered more accurately than by injection, resulting in fewer large swings in the blood sugar level, which makes the delivery of insulin easier, makes the food regime more comfortable, reduces sharp, low blood glucose level and eliminates unpredictable effects of intermediate or long acting insulin [9].

As glucose level increases in the patient's blood, the pancreas responds to this increase by producing an appropriate amount of insulin to consume glucose in the blood and reduce it to the normal level [5]. This process can be modelled and simulated if we develop a formula that can determine the relationship between real-time glucose level and insulin dose. To achieve this goal, different parameters in the human body should be considered.

This paper proposes a new methodology for modeling human body responsiveness to glucose intake and insulin injection using artificial neural networks (ANNs). The paper shows a neural network-based model that determines the relation between different human factors, the level of blood sugar and the appropriate insulin dose needed. In this paper, ten parameters (patient's gender, patient's age, body mass index, previous total daily insulin dose, history of the disease, smoking factor, family history, diabetic type, creatinine clearance and accumulative glucose test) are taken into consideration to investigate the relationship between them and the glucose and insulin levels.

2. Previous Work

Many researchers have worked in this field. Here, we overview some research papers that have discussed the blood glucose control process in order to predict the right amount of insulin dose. In [9], the researchers have aimed mainly at determining the appropriate insulin dose for diabetic patients automatically based on patient's historical data in real time. A Markov process was used for modeling blood sugar level, which can be used to determine the future value for a random variable depending on its history through the current observation. An experiment was conducted in Jordanian hospitals. Four factors were taken into consideration; the body weight, the amount of carbohydrate in the breakfast meal, the amount of carbohydrate in the lunch meal and the amount of carbohydrate in the dinner meal. For the body weight, three weights were considered; 100 lb, 200 lb and 300 lb. One patient was chosen for each category and for 27 days, the amount of carbohydrate in the breakfast and dinner was changed in three levels; 30, 60 and 120 grams, while the amount of carbohydrate in the lunch meal was changed in three levels; 60, 120 and 180 grams. The problem of this method is that the prediction of blood sugar values is depending on a few factors, which are the amount of carbohydrate and the body weight, which will not give an accurate prediction, because there are other factors that must be considered.

In [10], researchers developed a neural network algorithm to adjust the appropriate next insulin dose based on the history of blood-glucose measurement and insulin dose setting. 25000 data recorded from 747 insulin-pump users were used to achieve a generalization. An insulin pump device was designed and controlled by a neural network. The researchers used the neural network technology to predict the insulin dose which we will use for the same goal, but in their model, they depended on the history of blood-glucose measurement, while our model will take other parameters as well as the future value of blood-glucose measurement into account.

In [11], the researchers presented a self-tuning algorithm to adjust an on-line insulin dosage in Type 1 diabetic patients. This dosage doesn't need information about insulin-glucose dynamics. In this model, three daily doses were programmed, where two types of insulin were used: rapid and slow. The results of a closed loop simulation were illustrated by a nonlinear model of the subcutaneous insulin-glucose dynamics with meal intake in diabetes Type 1 patients. This model is not safe and doesn't give true results, because it predicts the value of the insulin dose depending on a nonlinear insulin-glucose model which is totally dependent on the meal.

In [12], a microcomputer program was developed to use educated and assisted information whenever diabetes patients needed to make a decision in conjunction with self-monitoring of blood glucose. This information consists of the amount of optimum insulin dosage and the time at which this dosage should be taken. This information was useful and objective according to the researchers' opinion because it can be obtained from insulin sensitivity and is mathematically substantiated, in addition to that good control of blood glucose was achieved.

In [13], the researchers studied the application of neural networks for modeling glucose level in diabetic patients' blood. Recurrent neural network and time neural network were compared to linear model and nonlinear compartment model. The experiment showed that taking the proper error in consideration improved the results. A powerful model was achieved by combination of linear error model and recurrent neural network and gave the best results for blood glucose prediction.

In [5], blood glucose metabolism was studied to predict the glucose concentration using offline training for artificial neural network model. The prediction was based on accessible information, like physical effort of the patients, food intake and blood glucose readouts. The study performed online prediction using a special particle filter. This study discussed the level of glucose in the blood. The difference between this study and our model is that our model uses more factors and predicts and determines more accurately the proper insulin dose for the patient in addition to the glucose value.

In [24], the study proposed a Type 1 diabetes glucose-insulin regulator using an artificial high-order recurrent neural network. Using this network, a nonlinear system will be identified and controlled in order to represent the pancreas behavior for diabetic patients. This model uses Kalman filter algorithm to get a quick conversance and uses safety block between the output control system and the patients. This model uses a feed forward neural network to control the glucose values in Type 1 diabetic patients' blood. It doesn't consider any parameters related to the patient or to the disease, except the glucose readouts. Further, it doesn't include Type 2 diabetics in the study and the insulin dose is not considered.

In [25], the paper presented two models to simulate the glucose-insulin interaction for Type 1 diabetes children only. The models were based on a combination of Compartmental Models (CMs) and artificial Neural Networks (NNs). The database used consists of a continuous glucose monitoring, insulin dosages and food intake. The system provided short-term prediction of glucose values. The paper presents a prediction system for glucose-insulin metabolism for children with Type 1 diabetes. It takes only three parameters into account and doesn't determine the proper insulin dose for the prediction of glucose values. Although it uses continuous data about glucose-insulin readouts, it doesn't predict any insulin dosages. In [14], the researchers presented an automatic blood glucose classifier to help the specialist provide a better interpretation for blood glucose readouts in case of gestational diabetes. Their paper compared six different feature selection methods for two learning methods; decision tree and neural networks. Three searching algorithms (Genetic, Greedy and Beat First) were companied with two evaluators (Wrapper and CSF). The best results were obtained when the model consists of decision tree with a feature set selection with Wrapper evaluator and Best First search algorithm. In spite of the results, the goal was to provide a classification system and not to predict a future value or determine insulin dosages.

In addition to the mentioned previous work, literature has many other contributors in this filed. Some other relevant publications can be found in [26]-[30]. In [26]-[28], blood glucose was predicted using artificial neural networks trained with the AIDA diabetes simulator. In [29], the goal was to find technological solutions to manage and treat diabetes. In [30], a neuro-fuzzy system was studied in order to improve diabetes therapy.

3. METHODOLOGY

3.1 Data Collection

In this paper, data and parameters were collected from 149 patients who have diabetes mellitus in normal conditions and are experiencing normal diet. These patients were using sliding a scale system to measure their sugar level. A medical record was created for every case of them. Some information was taken from the patient's file in the hospital, while other information was taken from the patient himself. The 149 patients who were taking insulin injection in the abdomen area. were previewed starting from May to September 2104 in Jordanian hospitals. These hospitals are: Princess Basma Teaching Hospital, King Abdullah University Hospital and Jordan University Hospital. Table 1 shows the parameters collected from the patients and used in this study.

Table 1. Parameters used in the study.

#	Name	Description
1	Patient's gender	Determines whether the patient is male or female.
2	Patient's age	Determines the age of the patient.
3	Body mass index (BMI)	Calculated by the formula (M/L^2); where M is the mass in kg and L is the height in meters; the normal range for body mass index is 18-24.
4	Previous total daily insulin dose (TDID)	It is the total insulin dose that patient used to take at home throughout the day.
5	The history of the disease	How long the patient had diabetes.
6	Smoking factor	Determines whether the patient is a smoker or a non-smoker.
7	Family history (genetic factor)	Determines whether the genetic factor exists or not.
8	Diabetic type	Determines whether the patient suffers from Type 1 or Type 2 diabetes.
9	Creatinine Clearance (CC)	It is a sign of efficiency of the kidney work, calculated from age, gender, weight and creatinine value in the blood.
10	Accumulative glucose test (HbA1C)	It is a test to determine the glucose accumulative average in the blood within the last three months.

3.2 Neural Network

In this paper, different neural network architectures have been implemented and studied to decide which one is the best. The parameters which were previously explained will be normalized and used as inputs to the network together with the blood sugar level of the patient. The desired output is the insulin dose ranging from 140 to 180 mg/dL [10]-[11]. The network uses the relation between all the inputs and the target to determine and adjust the weights of the connections to get a zero difference between the actual and the desired output in the training phase. The data is divided into three parts; training data (70%), testing data (15%) and validation data (15%) [12]-[15].

Considering the parameters which have been previously explained and in order to create a trained neural network, we need to provide the network with maximum number of diabetes patients' information. Figure 2 shows a basic block diagram for the neural network that is going to be used to determine the proper insulin doses based on the patients' data. The input to the neural network is the medical profile for the patient which was previously created and prepared. Because of the nature of the input data, it needs to be prepared before being used in the network [15]-[16].

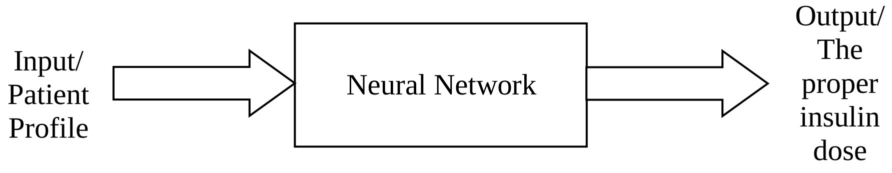


Figure 2. Neural network block diagram.

This preparation process includes several operations for every parameter, which will be explained later. Figure 3 shows the steps of modeling the neural network. First step is to arrange the input with its corresponding output. Then, the input profiles are prepared (quantification, normalization).

The data of the patient will be divided based on the level of glucose respective to insulin dose reduction. For this goal, a new indicator will be added to show whether the current insulin dose reduces the glucose to normal level or not. The input data doses that reduce the glucose to normal level are also divided into two parts; the first part used to training and the second part used in testing. The doses that didn't reduce the glucose to the normal level are assumed to be improper doses and will be used in the validation part to get actual outputs that represent the proper doses for the used inputs [17]-[22].

3.3 Parameters' Correlation

In this paper, the correlation between the patient input parameters and the glucose level is calculated. The correlation indicates the effect of the parameters on the diabetes mellitus. The correlation between the input parameters and the insulin doses is calculated as well. It shows the effect of the input parameters on the insulin doses. Table 2 shows the average glucose and insulin dose for both genders in the cases under consideration. The table shows an increase in both averages in the males' cases. Table 3 shows the average glucose and insulin doses for both smokers' and non-smokers' cases. It is clear that both averages are higher in the smokers' cases. Table 4 shows the average glucose and insulin doses for genetic and non-genetic cases. It is clear that both averages are higher in the genetic cases. Table 5 shows the average glucose and insulin doses for both genders for Type 1 and Type 2 diabetes. It is clear that the averages are higher in the Type 1 cases.

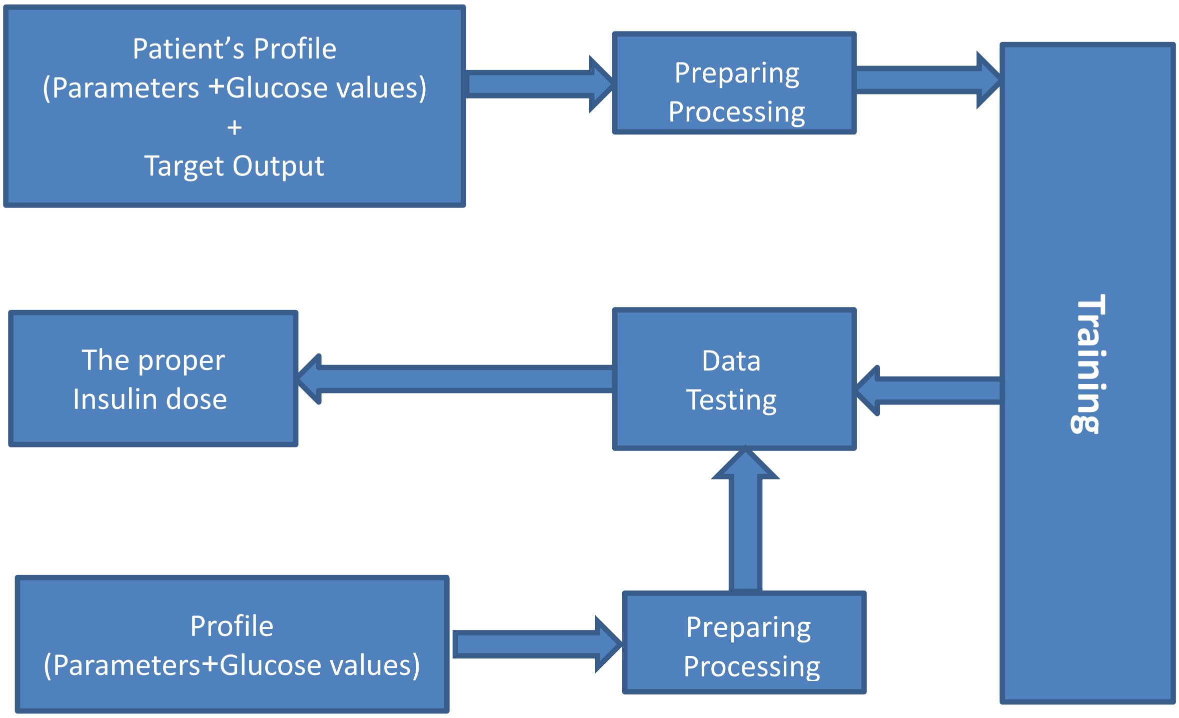


Figure 3. Block diagram for creating the neural network model [23].

Table 2. Average glucose/insulin for males and females [23].

Gender Parameter	Males	Females
Average Glucose (mg/dL)	214.8	264.9
Average Insulin (unit)	5.4	10

Table 3. Average glucose/insulin for smokers and non-smokers [23].

Smoking Factor	Smokers (50 Cases)	Non-smokers (85 Cases)
Average Glucose (mg/dL)	243.9	226
Average Insulin (unit)	7.9	6.5

The correlation factor can be calculated using the following equation [19]-[20]:

$$Corr = \frac{\frac{1}{n}\sum_{i=1}^{n}((Y_i - \mu_Y)(T_i - \mu_T))}{\sigma_Y \sigma_T} \tag{1}$$

where, Corr: correlation factor, n: number of samples, Y: the predicted values, T: the actual values
μ_{Y}, μ_{T} : the mean value of predicted values and actual values, respectively.
σ_Y, σ_T: the standard deviation of predicted values and actual values, respectively, which can be calculated using:

$$\sigma_Y = \sqrt{\frac{(Y-Y')^2}{n}}, \sigma_T = \sqrt{\frac{(T-T')^2}{n}} \tag{2}$$

Table 4. Average glucose/insulin for genetic and non-genetic cases [23].

Genetic Factor	Genetic (69 Cases)	Non-genetic (55 Cases)
Average Glucose (mg/dL)	247	223.5
Average Insulin (unit)	7.5	7.2

Table 5. Average glucose/insulin for Type 1 and Type 2 diabetes [23].

Diabetes type Parameter	Type 1 (20 Cases)	Type 2 (109 Cases)
Average Glucose (mg/dL)	262.3	232.3
Average Insulin (unit)	8.6	7.1

Table 6 shows the correlation of each parameter with the insulin dose in descending order. From the table, we can see that accumulative glucose (HbA1C) has the highest correlation, which comes from the fact that the insulin intake is highly correlated to the glucose level (correlation= 0.84877), which is in turn related to the accumulative glucose. The HbA1C parameter is considered one of the most important parameters to determine the insulin dose.

Table 6. Correlation between patient profile parameters and insulin [23].

Parameter	Correlation
HbA1C	0.6102
TDID	0.3167
Gender	-0.3109
History	0.2478
Smoking Factor	0.0963
Type	-0.0658
Age	-0.0400
Genesis	0.0204
CC	-0.0086
BMI	-0.0041

Table 7 shows the correlation of each parameter with the glucose level in descending order. From the table, we can see that accumulative glucose (HbA1C) has the highest correlation as well, which comes from same reason that the accumulative glucose is related to the glucose level in the blood.

3.4 Neural Network Inputs

The inputs for the neural network are the parameters that were previously discussed, in addition to glucose readouts and a matrix called (**Per**) containing four sub-matrices, (**P1**, **P2**, **P3** and **P4**), where:

- **P1**= all insulin-glucose readouts for all patients given at 5:00 am.

Table 7. Correlation between patient profile parameters and glucose [23].

Parameter	Correlation
HbA1C	0.6995
TDID	0.3274
History	0.2440
Gender	0.2242
Genetic Factor	-0.1069
Type	0.0859
Smoking Factor	0.0803
Age	-0.0465
CC	-0.0259
BMI	-0.0075

- **P2**= all insulin-glucose readouts for all patients given at 11:00 am.
- **P3**= all insulin-glucose readouts for all patients given at 5:00 pm.
- **P4**= all insulin-glucose readouts for all patients given at 11:00 pm.

Each sub-matrix contains 5 columns (variables) as follows:

- Column number 1: contains all the glucose readouts for all the patients which were taken in that period.
- Column number 2: corresponding insulin doses (the target).
- Column number 3: the difference between the glucose current readout and the next readout to distinguish whether the dose is correctly working or not.
- Column number 4: contains a factor to determine whether or not the patient goes to the healthy glucose level after he was given an insulin dose. If the insulin dose reduces the glucose to the normal level, the factor is (+1), while if it failed to reduce the glucose to the normal level, then it is (-1).
- Column number 5: period indicator, to determine the time for this dose.

First, periods were independently discussed to distinguish whether the time of the insulin dose is an effective parameter or not. The patient's response to insulin doses was taken into consideration to find out whether or not it could be changed according to the dose time. To determine the period's effect, the correlation between the glucose and the insulin doses in each period was measured in normal cases (in which insulin doses reduce the glucose to the normal level). Table 8 shows the correlation between the glucose values and the insulin doses in each time period and the number of samples in each period.

Because there are no obvious differences between period correlations and because the number of samples is small, time factor was not considered as a parameter. Figure 4 shows the architecture of the neural network with its all inputs. Inputs from 4 to 13 were arranged based on Table 6 and Table 7. The inputs of the neural network are:

- Input number 1: glucose readouts; included in the **Per** matrix.
- Input number 2: glucose difference between current and next readouts; included in the **Per** matrix.
- Input number 3: normal or abnormal glucose range; included in the **Per** matrix.
- Input number 4: HbA1C, referred to as **O_Mat**.
- Input number 5: TDID, referred to as **F_Mat**.
- Input number 6: History, referred to as **H_Mat**.
- Input number 7: Gender, referred to as **A_Mat**.
- Input number 8: Genetic Factor, referred to as **J_Mat**.

- Input number 9: Type, referred to as **K_Mat**.
- Input number 10: Smoking Factor, referred to as **I_Mat**.
- Input number 11: CC, referred to as **L_Mat**.
- Input number 12: Age, referred to as **B_Mat**.
- Input number 13: BMI, referred to as **E_Mat**.

Table 8. Correlation and number of normal samples in each time periods [23].

Periods	Insulin-glucose correlation	Number of samples
All periods	0.85	228
P1 (5:00 am)	0.82	61
P2 (11:00 am)	0.82	62
P3 (5:00 pm)	0.87	62
P4 (11:00 pm)	0.92	43

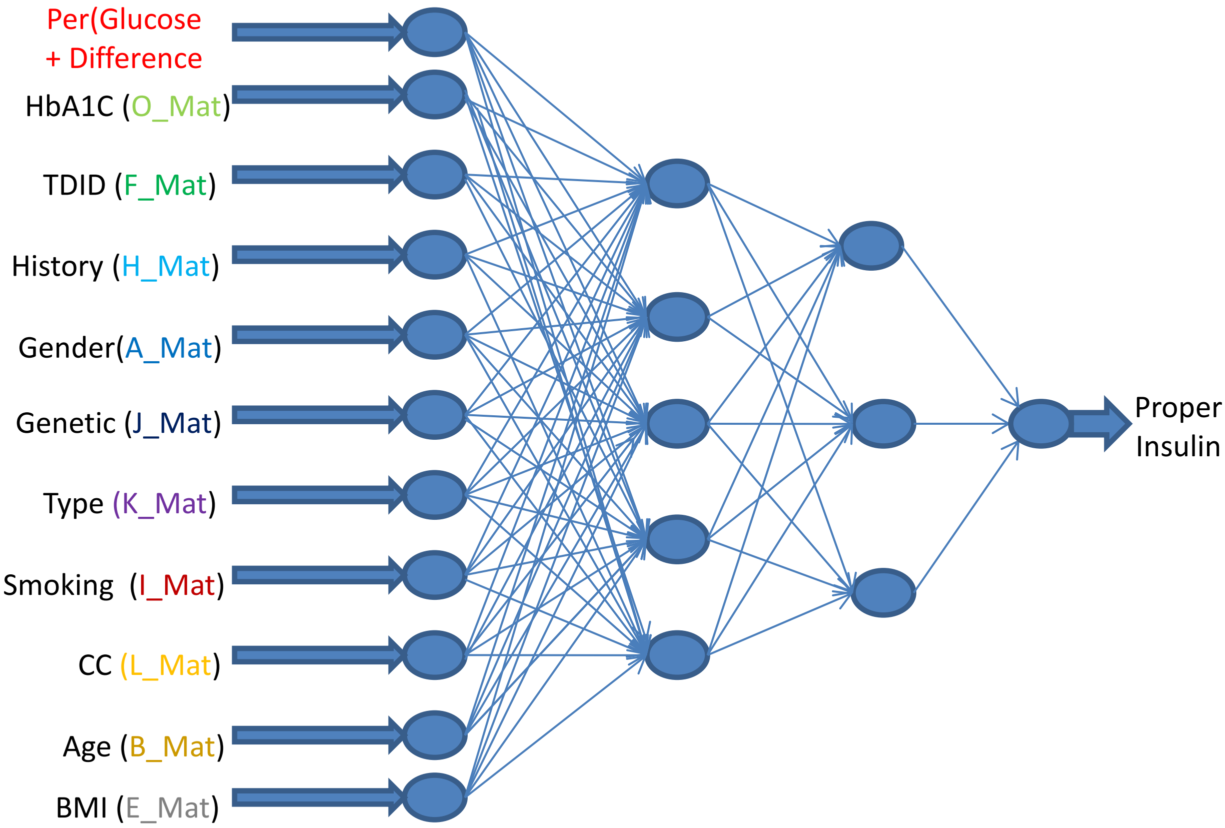


Figure 4. Data presentation in the neural network [23].

4. Results and Discussion

In this paper, several scenarios of different neural network models have been implemented and tested with different combinations of inputs. The main goal was to find the best model/input combination that would give the best insulin dose. In order to achieve our goal, one hidden layer and two hidden layer architectures will be investigated. The number of hidden layers will be referred to as HL. The number of neurons in each hidden layer will be varied and referred to as NN. This factor will be changed; the initial value of this factor will be (2) and it will be increased until it becomes equal to the number of inputs (which will be referred to as N). In the case of one hidden layer, the number of neurons for this hidden layer will be equal to NN, while in the case of two hidden layers, the number of neurons of the first hidden layer will be equal to 2*NN, while the number of neurons of the second hidden layer will be NN. Different neural network combinations and scenarios will be investigated using Matlab in order to determine the best model that has the best overall results. The mean factor to compare the scenarios is the Mean Square Error (MSE), which can be calculated by [19]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - T_i)^2 \tag{3}$$

4.1 One Hidden Layer-Architecture Results

In this part, a one hidden layer neural network has been implemented. The input size (N) was changed from 4 to 13. In the case when the input size was four, the four inputs were (Input number 1 to Input number 4). In the case when the input size was five, the five inputs were (Input number 1 to Input number 5),… and so on until we reach the input size of 13, where all inputs from Input number 1 to Input number 13 have been used. For every input case, the number of neurons (NN) in the hidden layer was changed from 2 to N. This will lead to 75 scenarios of different input sizes and different numbers of neurons in the hidden layer. The scenarios were labeled as SN, as shown in Table 9. Table 9 shows the results of one hidden layer neural network. Scenario number 29 was the best among all the scenarios. The input size (N) was 7 and the number of neurons (NN) was 6. The table shows the results of the

Table 9. Results for *one* hidden layer, NN= 2 to N (normal cases) [23].

SN	N	NN	Corr.	MSE	Training Time (s)	SN	N	NN	Corr.	MSE	Training Time (s)
1	4	2	0.8562	8.4683	1.99	72	10	7	0.8657	12.7573	5.47
2	4	3	0.8558	8.4847	2.11	73	10	8	0.8286	16.1753	2.51
3	4	4	0.8557	8.4858	2.36	74	10	9	0.7983	15.3577	3.27
7	5	2	0.9094	6.9048	1.57	75	10	10	0.8456	15.8539	4.02
8	5	3	0.9077	7.005	2.59	85	11	2	0.8342	9.5567	2
9	5	4	0.9032	6.9725	2.61	86	11	3	0.8333	10.9242	1.9
10	5	5	0.9035	7.2051	3.03	87	11	4	0.8328	12.6808	2.48
15	6	2	0.8969	8.0996	2.08	88	11	5	0.5431	23.5971	3.4
16	6	3	0.8818	9.5068	2.43	89	11	6	0.6105	26.0511	8.25
17	6	4	0.8983	7.6662	3.6	90	11	7	0.4476	33.071	3.42
18	6	5	0.781	16.4711	2.37	91	11	8	0.4719	29.3541	3.89
19	6	6	0.7424	20.8346	3.83	92	11	9	0.5803	21.3855	5.26
25	7	2	0.8809	9.4062	2.74	93	11	10	0.4521	51.2194	5.73
26	7	3	0.9097	7.2851	2.04	94	11	11	0.4987	33.1392	7.25
27	7	4	0.8629	12.2306	2.6	105	12	2	0.8157	10.7657	1.83
28	7	5	0.8864	9.5751	3.25	106	12	3	0.8116	12.1597	3.01
29	**7**	**6**	**0.9315**	**5.4135**	**2.75**	107	12	4	0.83	14.1061	1.94
30	7	7	0.9217	5.9918	3.69	108	12	5	0.7931	13.4011	6.36
37	8	2	0.9223	5.953	1.97	109	12	6	0.5103	23.909	3.83
38	8	3	0.891	8.2438	2.16	110	12	7	0.4282	32.1775	3.7
39	8	4	0.8674	10.1898	3.59	111	12	8	0.4243	30.5901	4.86
40	8	5	0.8804	9.4356	2.83	112	12	9	0.635	21.3933	4.75
41	8	6	0.8745	11.5904	2.71	113	12	10	0.48	32.069	6.75
42	8	7	0.9079	6.9399	3.22	114	12	11	0.502	32.6276	9.07
43	8	8	0.8231	19.9454	2.82	115	12	12	0.5226	38.6858	12.41
51	9	2	0.9083	7.1136	2.23	127	13	2	0.8508	8.1012	3.08
52	9	3	0.8896	8.5694	2.08	128	13	3	0.8203	15.2744	2.58
53	9	4	0.8961	8.0417	3.04	129	13	4	0.5097	29.1451	2.46
54	9	5	0.8604	13.1336	3.65	130	13	5	0.5334	22.8252	3.43
55	9	6	0.8929	9.1926	3.59	131	13	6	0.515	24.146	2.96
56	9	7	0.8817	10.0915	3.09	132	13	7	0.4636	34.2889	3.77
57	9	8	0.822	12.8006	5.02	133	13	8	0.4926	30.04	3.7
58	9	9	0.823	12.8966	3.71	134	13	9	0.513	30.7419	8.26
67	10	2	0.8832	9.4369	2.48	135	13	10	0.5221	34.7157	11.34
68	10	3	0.8605	11.3827	3.07	136	13	11	0.5204	32.9236	9.23
69	10	4	0.8439	15.12	2.14	137	13	12	0.5328	28.7386	10.66
70	10	5	0.8934	8.6533	2.16	138	13	13	0.5085	28.2032	10.84
71	10	6	0.8672	13.011	9.29						

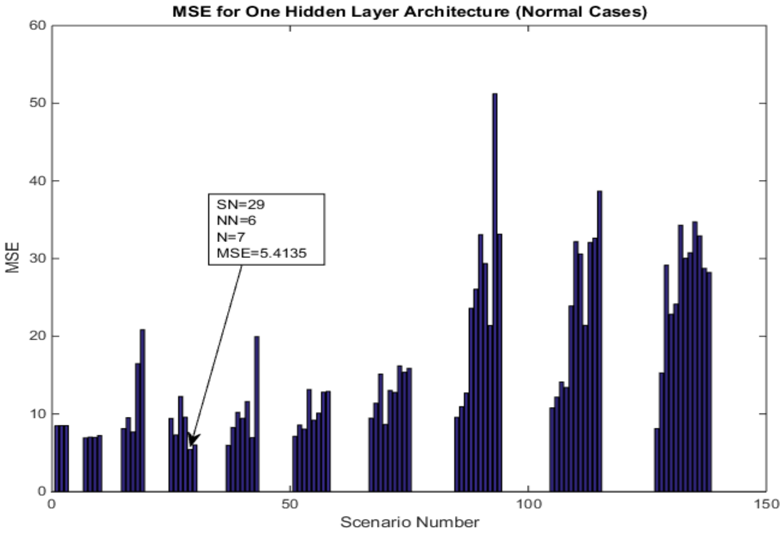


Figure 5. One hidden layer scenarios *vs.* MSE (normal cases) [23].

Table 10. Results for *one* hidden layer, NN= 2 to N (abnormal cases) [23].

SN	N	NN	Corr	MSE	SN	N	NN	Corr	MSE
1	4	2	-0.7011	415.055	72	10	7	-0.2038	130.412
2	4	3	-0.7088	249.964	73	10	8	-0.1083	279.252
3	4	4	-0.7114	214.299	74	10	9	-0.3052	466.544
7	5	2	-0.0173	127.864	75	10	10	0.0618	214.826
8	5	3	-0.023	128.49	85	11	2	0.5667	270.886
9	5	4	-0.1774	138.771	86	11	3	0.5024	127.568
10	5	5	-0.1504	131.275	87	11	4	0.0392	252.909
15	6	2	-0.1663	129.429	88	11	5	-0.243	352.265
16	6	3	-0.0343	127.966	89	11	6	0.1349	299.206
17	6	4	-0.2809	372.335	90	11	7	0.232	3230.6
18	6	5	0.3146	1738.44	91	11	8	0.1042	1397.1
19	6	6	0.3502	2129.36	92	11	9	-0.0332	780.922
25	7	2	0.08091	663.815	93	11	10	0.1356	425.308
26	7	3	-0.2346	165.87	94	11	11	0.4511	364.769
27	7	4	-0.1129	467.074	105	12	2	0.2111	301.557
28	7	5	-0.2888	293.282	106	12	3	0.3227	118.661
29	7	6	-0.5796	311.631	107	12	4	0.2158	193.543
30	7	7	-0.2927	230.686	108	12	5	-0.4071	169.868
37	8	2	0.1638	269.832	109	12	6	-0.2415	255.226
38	8	3	-0.3729	133.849	110	12	7	0.036	287.44
39	8	4	-0.3006	603.954	111	12	8	-0.2248	325.014
40	8	5	-0.0467	149.393	112	12	9	0.0259	1788.97
41	8	6	-0.2008	196.793	113	12	10	0.0185	186.216
42	8	7	0.0521	111.905	114	12	11	-0.2651	174.921
43	8	8	0.2686	344.154	115	12	12	0.2148	165.216
51	9	2	0.5682	87.2875	127	13	2	0.6198	75.1694
52	9	3	-0.066	129.276	128	13	3	0.0798	211.182
53	9	4	-0.0909	130.158	129	13	4	0.5105	723.819
54	9	5	-0.1384	258.689	130	13	5	0.0748	111.795
55	9	6	-0.2429	239.904	131	13	6	0.1437	355.578
56	9	7	0.1439	369.6	132	13	7	-0.0059	800.93
57	9	8	-0.2437	668.028	133	13	8	0.0092	371.203
58	9	9	0.6097	607.261	134	13	9	0.1511	179.785
67	10	2	-0.2459	130.456	135	13	10	0.0482	230.17
68	10	3	-0.2401	129.275	136	13	11	0.0887	125.798
69	10	4	-0.2342	136.658	137	13	12	-0.0971	173.311
70	10	5	-0.195	170.287	138	13	13	-0.014	200.059
71	10	6	-0.2566	190.287					

normal cases, where the insulin dose that was given to the patient reduced the glucose level to the normal range. The table shows that the results of this scenario have high correlation between the estimated insulin dose and the actual insulin dose that was given to the patient. The table shows that the training time was very small in scenario 29 and in the other scenarios as well. Figure 5 shows the MSE results indicating the best scenario (SN=29) as well.

In Table 10, the results were presented for the abnormal cases, where the insulin dose that was given to the patient did not reduce the glucose level to the normal range. The table shows very high MSE and very low correlation values between our estimated insulin dose and the insulin dose that was given to the patient. This result was expected, because the insulin dose that was given to the patient was inaccurate and far from truth and our estimated insulin doses should not agree with it.

4.2 Two Hidden Layer-Architecture Results

In this part, a two hidden layer neural network has been implemented. The input size (N) was changed from 4 to 13. In the case when the input size was four, the four inputs were (Input number 1 to Input number 4). In the case when the input size was five, the five inputs were (Input number 1 to Input number 5),… and so on until we reach the input size of 13, where all inputs from Input number 1 to Input number 13 have been used. For every input case, the number of neurons (NN) in the hidden layer was changed from 2 to N. The number of neurons in the first hidden layer was 2N, while the number of neurons in the second hidden layer was N. This will lead to 75 scenarios of different input sizes and different numbers of neurons in the hidden layers. The scenarios were labeled as SN, as shown in Table 11. Table 11 shows the results of two hidden layer neural network. Scenario number 31 was the best among all the scenarios. The input size (N) was 7 (same as in scenario 29) and the number of neurons (NN) was 4 in the first hidden layer and 2 in the second hidden layer, totaling 6 neurons (same as in scenario 29). The table shows the results of the normal cases, where the insulin dose that was given to the patient reduced the glucose level to the normal range. The table shows that the results of this scenario have high correlation between the estimated insulin dose and the actual insulin dose that was given to the patient. The table shows that the training time was very small in scenario 31 and in the other scenarios as well. Figure 6 shows the MSE results indicating the best scenario (SN=31) as well. Figure 7 shows the MSE results of all the 150 scenarios.

In Table 12, the results were presented for the abnormal cases, where the insulin dose that was given to the patient did not reduce the glucose level to the normal range. The table shows very high MSE and very low correlation values between our estimated insulin dose and the insulin dose that was given to the patient. This result was expected, because the insulin dose that was given to the patient was inaccurate and far from truth and our estimated insulin dose should not agree with it. Figure 8 shows the MSE results of all the 150 scenarios.

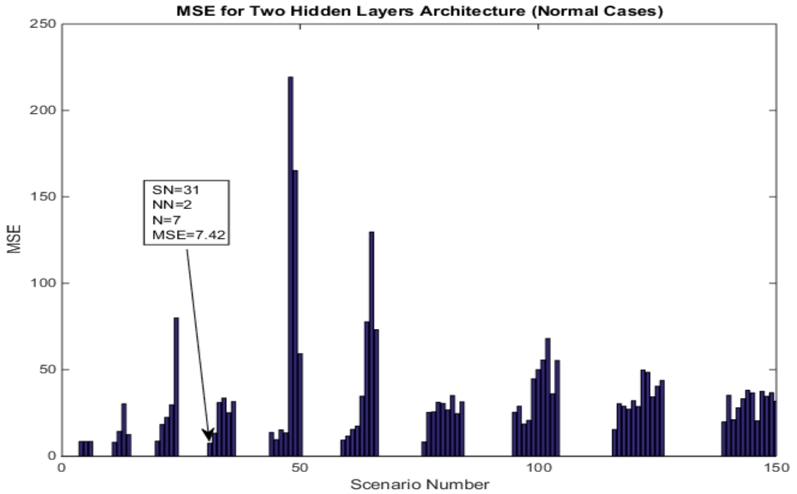


Figure 6. Two hidden layer scenarios *vs.* MSE (normal cases) [23].

Table 11. Results for *two* hidden layers, NN=2 to N (normal cases) [23].

SN	N	NN	Corr	MSE	Training Time	SN	N	NN	Corr	MSE	Training Time
4	4	2	0.857	8.4385	98.94	81	10	7	0.6958	26.7493	39.54
5	4	3	0.857	8.4376	139.48	82	10	8	0.634	35.0497	78.02
6	4	4	0.857	8.4371	94.11	83	10	9	0.7159	24.5227	108.71
11	5	2	0.8885	7.9654	3.6	84	10	10	0.6653	31.3797	126.58
12	5	3	0.7824	14.3083	5.78	95	11	2	0.6388	25.3486	2.82
13	5	4	0.6557	30.2161	10.68	96	11	3	0.6103	28.8664	3.09
14	5	5	0.8169	12.4147	11.8	97	11	4	0.763	18.5614	9.06
20	6	2	0.8885	8.6562	3.4	98	11	5	0.7239	20.6724	10.89
21	6	3	0.7806	18.259	4.58	99	11	6	0.3986	44.6389	37.18
22	6	4	0.732	22.425	7.18	100	11	7	0.5592	50.0105	49.95
23	6	5	0.6866	29.64	7.18	101	11	8	0.4823	55.6072	243.68
24	6	6	0.5938	79.912	23.53	102	11	9	0.5352	68.0122	74.95
31	**7**	**2**	**0.9077**	**7.42**	**3.11**	103	11	10	0.5769	36.0068	348.79
32	7	3	0.8604	13.2376	4.94	104	11	11	0.4616	55.3296	289.85
33	7	4	0.7024	31.0033	5.57	116	12	2	0.7575	15.3573	4.86
34	7	5	0.7548	33.5846	8.34	117	12	3	0.5743	30.3351	5.72
35	7	6	0.68	25.0394	16.99	118	12	4	0.5817	28.8664	5.62
36	7	7	0.6833	31.5006	25.24	119	12	5	0.5998	27.1308	15.21
44	8	2	0.8616	13.7006	2.43	120	12	6	0.589	32.0639	27.25
45	8	3	0.8912	9.5164	5.05	121	12	7	0.5796	28.6148	58.11
46	8	4	0.8481	15.1299	5.78	122	12	8	0.4509	49.7981	62.25
47	8	5	0.8543	13.4325	6.82	123	12	9	0.5171	48.4747	127.88
48	8	6	0.4644	219.2823	21.43	124	12	10	0.4706	34.2603	173.52
49	8	7	0.7182	165.1854	40.43	125	12	11	0.4928	40.4477	171.63
50	8	8	0.6936	59.254	64.49	126	12	12	0.5319	43.7363	153.93
59	9	2	0.8769	9.2321	3.2	139	13	2	0.7443	19.7896	2.73
60	9	3	0.858	11.5581	4.92	140	13	3	0.456	35.19	4.73
61	9	4	0.8613	15.3942	9.41	141	13	4	0.7469	20.9884	6.8
62	9	5	0.7575	17.368	15.49	142	13	5	0.7292	27.9412	26.23
63	9	6	0.7316	34.6058	14.04	143	13	6	0.7634	33.1865	21.69
64	9	7	0.569	77.6212	45.5	144	13	7	0.4733	38.0705	111.03
65	9	8	0.6752	129.6203	69.8	145	13	8	0.4926	36.5358	43.86
66	9	9	0.5392	73.1059	94.44	146	13	9	0.6869	20.3738	210.67
76	10	2	0.8998	8.1551	3.35	147	13	10	0.5859	37.4392	178.57
77	10	3	0.8318	25.2445	3.46	148	13	11	0.5532	34.5468	175.65
78	10	4	0.7051	25.569	8.66	149	13	12	0.5023	36.6599	114.03
79	10	5	0.7487	31.1132	7.49	150	13	13	0.485	31.7137	399.4
80	10	6	0.6379	30.4688	19.63						

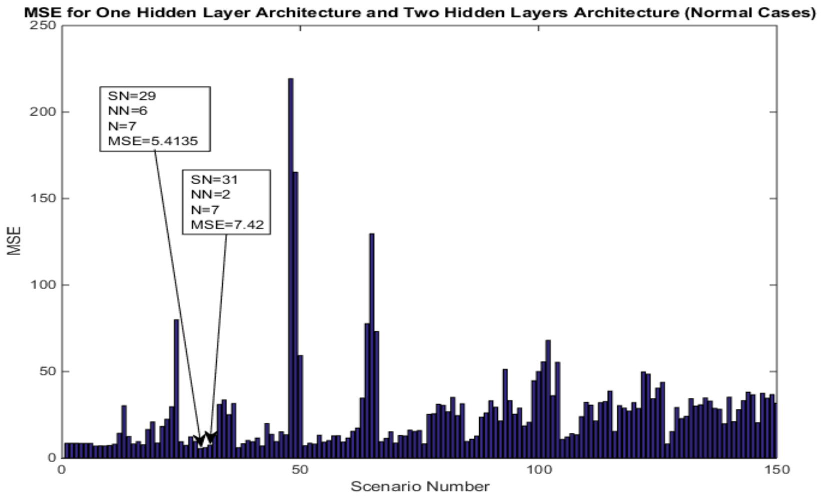


Figure 7. All scenarios *vs.* MSE (normal cases) [23].

Table 12. Results for *two* hidden layers, NN=2 to N (abnormal cases) [23].

SN	N	NN	Corr	MSE	SN	N	NN	Corr	MSE
4	4	2	0.7736	137.908	81	10	7	-0.0951	303.386
5	4	3	0.7668	145.385	82	10	8	0.3492	261.933
6	4	4	0.7627	149.726	83	10	9	0.1155	794.418
11	5	2	-0.0439	128.76	84	10	10	0.419	988.89
12	5	3	0.5536	1209.45	95	11	2	0.5643	73.2999
13	5	4	-0.3063	2628.33	96	11	3	0.5231	87.7208
14	5	5	0.1386	170.128	97	11	4	0.2985	1261.97
20	6	2	-0.1232	128.855	98	11	5	-0.138	1002.16
21	6	3	-0.0264	128.428	99	11	6	0.0628	327.401
22	6	4	0.5232	58.3101	100	11	7	0.0956	336.302
23	6	5	0.6149	3386.87	101	11	8	0.2405	259.946
24	6	6	-0.4645	12271.5	102	11	9	0.4853	361.524
31	7	2	0.219	250.472	103	11	10	0.1099	221.032
32	7	3	-0.2211	366.505	104	11	11	0.1657	441.659
33	7	4	0.0254	550.042	116	12	2	-0.0627	164.374
34	7	5	0.4985	715.249	117	12	3	0.2321	206.586
35	7	6	0.1834	797.093	118	12	4	0.4231	87.7208
36	7	7	-0.0719	147.863	119	12	5	0.1802	395.365
44	8	2	0.6417	209.077	120	12	6	0.3635	437.953
45	8	3	-0.4245	708.868	121	12	7	0.3707	158.253
46	8	4	-0.0693	469.126	122	12	8	0.0129	139.908
47	8	5	0.2435	398.054	123	12	9	0.1484	114.586
48	8	6	-0.2242	6194.88	124	12	10	0.1274	259.738
49	8	7	-0.1899	648.739	125	12	11	0.3391	167.358
50	8	8	0.1758	2159.4	126	12	12	0.281	141.162
59	9	2	-0.3566	288.816	139	13	2	0.1254	176.047
60	9	3	0.0608	3243.18	140	13	3	-0.4203	500.726
61	9	4	0.2364	117.433	141	13	4	0.411	2092.45
62	9	5	0.4535	96.9451	142	13	5	0.094	803.296
63	9	6	0.1131	4479.02	143	13	6	0.1014	256.486
64	9	7	-0.1486	653.451	144	13	7	0.0351	175.18
65	9	8	0.1135	3768.88	145	13	8	-0.03	146.857
66	9	9	0.6558	1395.86	146	13	9	0.3455	482.715
76	10	2	0.1486	105.647	147	13	10	0.6095	85.7366
77	10	3	0.2631	193.772	148	13	11	0.4303	87.258
78	10	4	0.105	550.041	149	13	12	0.1839	204.988
79	10	5	0.4139	258.184	150	13	13	0.42	157.063
80	10	6	0.1884	1759.14					

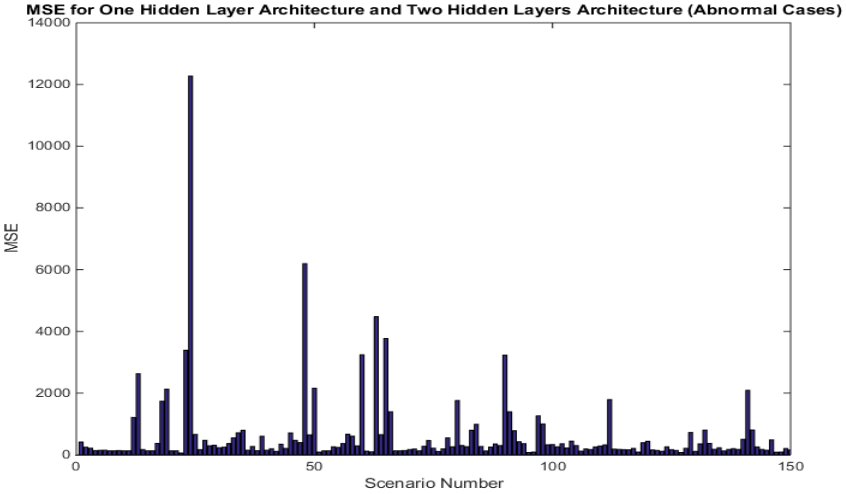


Figure 8. All scenarios *vs.* MSE (abnormal cases) [23].

Table 13. A comparison between proposed system and systems in literature.

Reference	Method	Number of Factors (Inputs)	RMSE (mg/dL)
[31]	Artificial Neural Networks (ANNs)	1	10, 18, 27
[32]	Recurrent Neural Networks (RNNs)	12	19.04
[33]	Convolutional Neural Networks (CNNs)	6	19.31, 29.3
Proposed	Artificial Neural Networks (ANNs)	11	2.3265

5. Summary and Conclusion

This paper aimed to discuss the effect of certain parameters on diabetes mellitus and on the insulin dose for diabetes patients, in order to determine the proper insulin dose for diabetic patients based on medical profiles based on neural networks. To determine the proper insulin dose, a neural network was modeled using glucose-insulin continuous readouts for in-hospital diabetic patients as input for our model, in addition to other parameters.

The parameters that were discussed in this paper are: patient's gender, patient's age, body mass index, previous total daily insulin dose, patient's nutrition status, history of the disease, smoking factor, family history, diabetic type, creatinine clearance and accumulative glucose test. The used samples were taken from three Jordanian hospitals, Princess Basma Teaching Hospital, King Abdullah University Hospital and Jordan University Hospital, from May to September 2014. The results show that the most effective parameter was the accumulative glucose, while the least effective parameter was the body mass index.

The results also show that the best architecture for our model was obtained when we used an architecture with one hidden layer, six neurons and seven inputs. The significant inputs were glucose readouts, glucose difference, normal range, accumulative glucose, history of the disease, total insulin dose and patient's gender. The MSE of the best model was 5.413 and the correlation was 0.9315 with negligible training time. Table 13 provides a comparison between our proposed methodology and other methods presented in literature [31]-[33]. It is clear that our proposed method has better performance in terms of RMSE. The RMSE of our proposed method was 2.3265, being larger than those for the other methods.

References

[1] Health Organization, [Online], Available: http://www.who.int/diabetes/en, accessed in Apr. 2019.

[2] World Health Organization–Diabetes Program, [Online], Available: http://www.World.who.int/diabetes /action_online/basics/en/index.html, accessed in Dec. 2014.

[3] Atlas, Diabetes, "International Diabetes Federation," [Online], Available: http://www.idf.org/diabetesatlas/5e/es/prologo, accessed in Dec. 2014.

[4] C. Klonoff, B. Buse, L. Nielsen, X Guan, L. Bowlus, H. Holcombe, E. Wintle and G. Maggs, "Exenatide Effects on Diabetes, Obesity, Cardiovascular Risk Factors and Hepatic Biomarkers in Patients with Type 2 Diabetes Treated for at Least 3 Years," Current Medical Research and Opinion, vol. 24, no. 1, pp. 275-286, Dec. 2007

[5] B. Thomas and V. Tresp, "A Nonlinear State Space Model for the Blood Glucose Metabolism of a Diabetic (Ein nichtlineares Zustandsraummodell für den Blutglukosemetabolismus eines Diabetikers)," at-Automatisierungstechnik, vol. 50, no. 5, pp. 228-236, Sept. 2009.

[6] S. Vashist, D. Zheng, K. Al-Rubeaan, J. Luong and F. Sheu, "Technology behind Commercial Devices for Blood Glucose Monitoring in Diabetes Management: A Review," Analytica Chimica Acta, vol. 703, no. 1, pp. 124–136, Jul. 2011.

[7] H. Park, K. Lee, H. Yoon and H. Nam, "Design of a Portable Urine Glucose Monitoring System for Health Care," Computers in Biology and Medicine, vol. 35, no. 4, pp. 275-286, Apr. 2004.

[8] American Diabetes Association, [Online], Available: http://www.diabetes.org/living-with-diabetes/treatment-and-care/medication/insulin/insulin-pumps.html, accessed in Apr. 2019.

[9] M. Otoom, H. Alshraideh, H. Almasaeid, D. López-de-Ipiña and José Bravo, "A Real-time Insulin Injection System," Proceedings of the Ambient Assisted Living and Active Aging- 5th International Work-Conference (IWAAL), pp. 120–127, Dec. 2013.

[10] A. Fidimahery and M. Milgram, "Applying Neural Networks to Adjust Insulin-pump Doses," Proceedings of the 7th IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing VII, pp. 182-188, Sept. 1997.

[11] R. DeFronzo, "Insulin Resistance, Lipotoxicity, Type 2 Diabetes and Atherosclerosis: The Missing Links. The Claude Bernard Lecture 2009," Diabetologia, vol. 53, no.7, pp. 1270-1287, Apr. 2010.

[12] T. Shimauchi, N. Kugai, N. Nagata and O. Takatani, "Microcomputer-aided Insulin Dose Determination in Intensified Conventional Insulin Therapy," IEEE Transactions on Biomedical Engineering, vol. 35, no. 2, pp. 167-171, Feb. 1988.

[13] T. Volker, T. Briegel and J. Moody, "Neural-network Models for the Blood Glucose Metabolism of a Diabetic," IEEE Transactions on Neural Networks, vol. 10, no. 5, pp. 1204-1213, Sept. 1999.

[14] E. Caballero-Ruiz, G. García-Sáez, M. Rigla, M. Balsells, B. Pons, M. Morillo, E. J. Gómez and M. E. Hernando, "Automatic Blood Glucose Classification for Gestational Diabetes with Feature Selection: Decision Trees *vs.* Neural Networks," Proc. of XIII Mediterranean Conference on Medical and Biological Engineering and Computing, pp. 1370-1373, Sept. 2013.

[15] M. Vasudev and J. Johnston, "Inpatient Management of Hyperglycemia and Diabetes," Clinical Diabetes", vol. 29, no. 1, pp. 3-9, Jan. 2011.

[16] A. Chennakesava, Fuzzy Logic and Neural Networks: Basic Concepts & Applications, India: New Age International, 2008.

[17] H. Simon, Neural Networks: A Comprehensive Foundation, New Jersey: Mc Millan, 2010.

[18] H. Demuth, M. Beale, O. De Jess and M. Hagan, Neural Network Design, 2nd Edition, USA: Martin Hagan, 2014.

[19] S. Milton and J. Arnold, Introduction to Probability and Statistics: Principles and Applications for Engineering and Computing Sciences, USA: McGraw-Hill Education, 2002.

[20] E. Lehmann and G. Casella, Theory of Point Estimation, Berlin: Springer Sci. & Bus. Media, 2006.

[21] Q. Wang, P. Molenaar, S. Harsh, K. Freeman, J. Xie, C. Gold, M. Rovine and J. Ulbrecht, "Personalized State-space Modeling of Glucose Dynamics for Type 1 Diabetes Using Continuously Monitored Glucose, Insulin Dose and Meal Intake: An Extended Kalman Filter Approach," Journal of Diabetes Science and Technology, vol. 8, no. 2, pp. 331-345, March 2014.

[22] S. Pappada, B. Cameron, P. Rosman, R. Bourey, T. Papadimos, W. Olorunto and M. Borst, "Neural Network-based Real-time Prediction of Glucose in Patients with Insulin-dependent Diabetes," Diabetes Technology & Therapeutics, vol. 13, no. 2, pp. 135-141, Feb. 2011.

[23] A. Bani-Younes, Modeling Human Body Responsiveness to Glucose Intake and Insulin Injection Using Neural Networks, Master Thesis, Jordan: Yarmouk University, 2014.

[24] O. Orozco, E. Castañeda, A. Rodrı́ıguez-Herrero, G. García-Saéz and M. Elena Hernando, "Glucose-Insulin Regulator for Type 1 Diabetes Using High-order Neural Networks," International Journal of Artificial Intelligence and Neural Networks (IJAINN), vol. 4, no. 3, pp. 40-47, Sept. 2014.

[25] S. Mougiakakou, A. Prountzou, D. Iliopoulou, K. Nikita, A. Vazeou and C. Bartsocas, "Neural Network Based Glucose-insulin Metabolism Models for Children with Type 1 Diabetes," Proceedings of the 28th IEEE EMBS Annual Int. Conf., New York City, USA, pp. 3545-3548, Aug. 30-Sept. 3, 2006.

[26] G. Robertson, E. D. Lehmann, W. A. Sandham and D. J. Hamilton, "Blood Glucose Prediction Using Artificial Neural Networks Trained with the AIDA Diabetes Simulator: A Proof-of-concept Pilot Study," Journal of Electrical and Computer Engineering, vol. 2011, Article ID 681786, pp. 1-11, 2011.

[27] W. A. Sandham, M. Z. Diaz, D. J. Hamilton, E. D. Lehmann, P. Tatti and J. Walsh, "Electrical and Computer Technology for Effective Diabetes Management and Treatment," Special Issue of the Journal of Electrical and Computer Engineering, 2011.

[28] W. A. Sandham, D. J. Hamilton, D. Nikoletou, C. MacGregor, A. Japp and K. Patterson, "Use of Artificial Neural Networks for Improved Diabetes Therapy," Proc. of Irish Signals and Systems Conference (ISSC-98), Dublin Institute of Technology, Dublin, Ireland, pp. 553-560, 25-26 June 1998.

[29] W. A. Sandham, D. Nikoletou, D. J. Hamilton, K. Paterson, A. Japp and C. MacGregor, "Blood Glucose Prediction for Diabetes Therapy Using a Recurrent Artificial Neural Network," Proc. of the IX European Signal Processing Conference (Eusipco-98), Island of Rhodes, Greece, pp. 673-676, 8-11 Sep. 1998.

[30] W. A. Sandham, D. J. Hamilton, A. Japp and K. Patterson, "Neural Network and Neuro-fuzzy Systems for Improving Diabetes Therapy," Proc. of the 20th Int. Conf. of the IEEE Eng. in Med. & Biol. Soc., Hong Kong Convention and Exhibition Centre, Hong Kong, vol. 20, Part 3/6, pp. 1438-1441, 1998.

[31] C. Pérez-Gandía, A. Facchinetti, G. Sparacino, C. Cobelli, E. J. Gómez, M. Rigla, A. de Leiva and M. E. Hernando, "Artificial Neural Network Algorithm for Online Glucose Prediction from Continuous Glucose Monitoring," Diabetes Technology & Therapeutics, vol. 12, no 1, pp.81-88, Jan. 2010.

[32] J.-W. Chen, , K. Li, P. Herrero, T. Zhu and P. Georgiou, "Dilated Recurrent Neural Network for Short-time Prediction of Glucose Concentration," Proc. of the 23rd European Conf. on Artificial Intelligence (IJCAI-ECAI), Int.l Workshop on Knowledge Discovery in Healthcare Data, pp. 69-73, 2018.

[33] K. Li, J. Daniels, C. Liu, P. Herrero-Vinas and P. Georgiou, "Convolutional Recurrent Neural Networks for Glucose Prediction," IEEE Journal of Biomedical and Health Informatics, DOI: 10.1109/JBHI.2019.2908488, Apr. 2019.

ملخص البحث:

يعدّ مرض السكري أحد أكثر الأمراض انتشاراً على مستوى العالم، وبخاصة في العالم الغربي حيث يكثر تناول الأطعمة السريعة وغير الصحية. وقد أجريت أبحاث متعددة لإيجاد طرق لتوقع مرض السكري وتشخيصه وعلاجه. ومن بين الطرق المستخدمة لهذا الغرض النمذجة الرياضية التي تستخدم لتطوير نماذج لجلكوز الدم وامتصاص الإنسولين.

في هذه الورقة، تم تطوير نموذج لتحديد جرعة الإنسولين المناسبة لمرضى السكري باستخدام شبكة عصبية اصطناعية. وقد تم تطوير النموذج مع أخذ عشرة متغيرات بعين الاعتبار (جنس المريض؛ عمر المريض؛ مؤشر كتلة جسم المريض؛ التاريخ المرضيّ؛ إجمالي جرعات الإنسولين اليومية؛ نوع مرض السكري؛ عامل التدخين؛ العامل الوراثي؛ تصفية الكرياتنين؛ الجلكوز التراكمي)، إضافة الى قراءات الجلكوز في الدم في الزمن الحقيقي. وجرى تطوير النموذج بناءً على مجموعة بيانات تعود الى 159 من مرضى السكري من 3 مستشفيات مختلفة.

وُجد أنّ أفضل النماذج من حيث الأداء هو ذلك النموذج المبني على طبقة مخفيّة واحدة مع ستّة عصبونات وسبعة مداخل. وكانت المداخل المهمة: قراءات الجلكوز؛ فرْق الجلكوز؛المدى الطبيعي؛ الجلكوز التراكمي؛ تاريخ المرض، الجرعة الكلية للإنسولين؛ عمر المريض. وبلغ متوسط مربع الخطأ للنموذج الأفضل (5.413)، بينما بلغ معامل الارتباط (0.9315) مع زمن تدريب يمكن إهماله.

THE IMPACT OF MOBILITY MODELS ON THE PERFORMANCE OF AUTHENTICATION SERVICES IN WIRELESS SENSOR NETWORKS

Iman Almomani[1] and Katrina Sundus[2]

(Received: 20-Jul.-2019, Revised: 3-Nov.-2019, Accepted: 30-Nov.-2019)

ABSTRACT

The applications of Wireless Sensor Networks (WSNs) are very important nowadays and could be found in many different life aspects. Broadcast authentication (BA) protocols are solutions to guarantee that commands and requests sent by the Base Station (BS), which controls the services provided by WSN, are authentic. Network mobility is considered one of the main challenges that WSN services in general and authentication protocols in particular are facing. Existing BA protocols did not give much attention to the effect of mobile BS or/and sensors on the behaviour of their protocols. Consequently, this paper provides a deep analysis of the impact of mobility on the performance of BA protocols. Three standard designs for BA protocols were studied in this research; Forwarding First (FF), Authentication First (AF) and Adaptive Window (AW). These three standard protocols were examined against four major mobility models. The results revealed that BA protocols behaved differently in terms of energy consumption and network delay with respect to mobility. For example, the delay in AW protocol was decreased by 47.6% in case of having fully mobile WSN; whereas the wasted energy was reduced by 37.5% in case of static BS and mobile sensors. Although the same authentication technique was applied in all three protocols, the mobility itself was a reason to enhance or degrade the performance of the authentication service which consequently affects the security of WSNs and their provided services. For example, when the BS was mobile and the sensor nodes were static, FF protocol decreased the delay by up to 98.81% compared to AF protocol and by up to 93.62% compared to AW protocol. On the other hand, AW Protocol saved the network energy by up to 94.49% compared to FF protocol and by up to 65.5% compared to AF protocol.

KEYWORDS

Authentication, Wireless sensor network (WSN), Security, Mobility, Broadcast authentication, Adaptive window protocol, Authentication first protocol, Forwarding first protocol, Internet of Things (IoT), Digital signature.

1. INTRODUCTION

Wireless sensor Network (WSN) is a group of spatially deployed sensor nodes that acknowledge or remotely observe diverse environmental variables or natural events [1]. WSN is currently practiced at various applications in both civilian and military fields [2]-[3]. Internet of Things (IoT) has become part of our daily life routines and its applications can be seen almost everywhere, such as cities [4]-[5], streets [6] and even universities [7]-[8]. One of the essential components of IoT environment is WSNs [1]. The node in a WSN is classified into a sensor node or a base station (BS) [10]. The sensor nodes are used to collect the surrounding natural events, process data, respond to BS requests and commands or transmit the data to other neighbour sensors. BS (also known as a sink node) mainly sends commands to the sensor nodes to perform particular tasks and receives the collected data from the sensor nodes to perform data aggregation and execute analysis on the collected data [9].

WSNs offer many attributes to encourage sensor deployment over IoT environments, such as low-cost deployment, decentralized nature, soft setting and tearing of the network, multi-hop communication transmission, as well as limited requested resources in terms of energy, processing, memory and communication bandwidth, appealing for more application areas. However, WSNs have more challenges than any other network type in respect to designing efficient security solutions [11]-[15].

Achieving security in WSN applications is very essential, especially in unattended environments and security monitoring applications [16]. Applying security mechanisms to WSN is quite challenging [8], [17]-[18]. This is due to the limited resources of sensor nodes, the nature of communication, the large

1. I. Almomani is with Department of Computer Science, Security Engineering Lab (http://sel.psu.edu.sa/) Prince Sultan University, Riyadh, KSA and with Department of Computer Science, University of Jordan, Amman, Jordan. Email: imomani@psu.edu.sa and i.momani@ju.edu.jo

2. K. Sundus is with Department of Computer Science, University of Jordan, Amman, Jordan. Email: sun.katrina@yahoo.com

and dense sensor node deployment and the dynamic topology of the network [18].

Wireless sensor nodes make WSNs an easy target to different types of attacks, including Denial of Service (DoS) attack due to the open wireless communication and physical risks [19]. To protect WSNs from DoS attacks, we need to ensure the authenticity of the transmitted messages. This could be achieved by applying some Broadcast Authentication (BA) mechanisms to decrease and contain the effect of DoS attacks.

BA is a growing subject in the field of WSN security. BA needs to handle the issue of transmitting messages while receiving other messages in a timely manner, especially in time-sensitive applications. Also, to consider the mobility effect in case of mobile networks [20] to ensure efficient broadcast authentication services. BA allows the BS to broadcast messages to all sensor nodes in the network in a secure manner. Several BA techniques have been proposed to secure WSNs [21]-[22]. However, the mobility of sensors and/or the BS was not taken into consideration. Mobility in WSNs could affect the performance of BA protocols significantly in terms of time delay and energy consumption.

Therefore, this paper investigates the impact of mobility on the behaviour of BAs. Three standard BA protocols; Forwarding First Protocol (FFP), Authentication First Protocol (AFP) and Adaptive Window Protocol (AWP) were implemented and examined against different mobility models. Moreover, FFP, AFP and AWP were evaluated using four main metrics: consumed energy, end-to-end delay, speed and pause time in the presence of different attack intensities. Additionally, four different mobility models were applied to the experimental environment of WSN to test thoroughly the performance of the three BA protocols: fully static WSN, dynamic sensors with static BS, static sensors with mobile BS and fully mobile WSN. The digital signature technique was chosen as a proof of authenticity for the messages transmitted over the network to be able to calculate the processing cost and the amount of consumed energy.

This paper is organized as follows: Section two presents a literature review. Section three summarizes the studied BA protocols and introduces the proposed system architecture. Section four presents the simulation environment, evaluation metrics and the attacking model. Section five presents the simulation results and analysis. Section six draws the conclusion and suggests recommendations for future tasks.

2. LITERATURE REVIEW

WSN is a type of wireless network that consists of a large number of resource-constrained sensor nodes and a small number of powerful devices called BS; collaborating together to accomplish a common task by communicating with each other via wireless links [23]. The BS transmits commands or requests to the sensor nodes which could be sent authenticated in some sensitive applications. In general, the security approaches require a certain amount of resources in order to be functional, including data memory, code space and energy to power the sensor nodes [24]. Therefore, the traditional security mechanisms with high computation and communication requirements are undesirable in WSNs and make achieving security a challenging task.

Many security mechanisms in the literature were proposed to protect WSNs against different types of attacks. Patil et al. [18] summarized different existing authentication techniques for WSNs with the main challenges that they are facing in such type of networks. In the following paragraphs, several approaches proposed to achieve broadcast authentication in WSN are discussed.

Timed Efficient Stream Loss-tolerant Authentication (TESLA) and its versions [25]-[26] as well as Digital Signature [27]-[29] techniques were used to implement the BA in WSNs. Furthermore, both techniques protect the entire network from different types of security attacks which assimilate an important role for achieving more trusted messages. In general, the security mechanisms that provide BA in WSNs can be classified into three main categories: Intrusion Prevention-based Systems (IPSs) [23], Intrusion Detection-based Systems (IDSs) [30]-[31] and a combination of both as Intrusion Prevention Detection-based Systems (IPDSs) [32]-[33]. Mittal in [17] summarized the most IDS community that is suitable for WSNs.

Han et al. [34] proposed a key agreement-based authentication technique for dynamic WSNs to decrease the overhead of the authentication process. However, the authors did not provide any simulation experiments to show the efficiency of their proposed approach.

Maidhili et al. [35] proposed an identity-based multi-user BA scheme to provide message authentication. The idea was to minimize the transmission rate to save energy. Moreover, specific authentication techniques were chosen to reduce the computation, but without considering the network mobility.

In [36], the authors proposed BA scheme for smart home. This scheme was based on Elliptic Curve Digital Signature Algorithm (ECDSA) to provide authentication of alarm messages or update messages of service providers in smart home environment. The purpose was to prevent attackers from accessing the home network and injecting forge messages. Another approach which was also based on ECDSA was presented in [37]. This approach supported multiuser BA to preserve both user's privacy and untracking. Both approaches did not consider or evaluate mobility in their proposals.

Shim in [38] proposed an ID-based multiuser BA scheme to minimize computation and communication costs of authentication services in WSNs. The focus was to test the proposed scheme on different hardware platforms, such as MICAz and Tmote Sky, used in real-life deployments. Mobility was not among their evaluation metrics and its effect was not examined.

The work in [39] deployed RSA-like public key cryptography to design a mechanism for multiuser BA in WSNs. The quantitative analyses that the authors conducted showed that their scheme was efficient in terms of storage and computational overheads. But again, there was no consideration for mobility models in their experimental environment.

A bidirectional BA scheme based on Merkle hash tree and TESLA protocol was proposed in [40]. The main idea was adding a verify node in the Merkle hash tree broadcast authentication. This node was responsible to store the entire hash tree. Consequently, their scheme reduced the transmission overhead and ensured secure communications between the central node and the leaf node. Although storage, communication and computation costs were considered in their evaluation and comparison metrics, but mobility effect was also absent in this research.

Applying security techniques in WSNs to achieve message authentication forces the sensor node to perform local operations inside each sensor to verify the correctness of the message, which costs the sensor some of its energy. However, WSN mobility could introduce more overhead on the sensor nodes and could affect the BA protocol performance.

As can be observed from the discussed literature, the current solutions did not highlight how mobility is affecting the performance of authentication services. Therefore, this study investigates the impact of mobility and illustrates to what extent it could affect the performance of broadcast authentication protocols in WSNs.

3. AUTHENTICATION PROTOCOLS

This section reviews the three BA protocols studied in this research, in addition to a brief overview about the mobility models in WSNs.

3.1 Forwarding First Protocol (FFP)

In FFP protocol [41], the BS sends digitally signed messages. Once the sensors receive these messages, the sensors will forward the messages immediately to the neighbour nodes before checking their validity. In other words, the messages will be forwarded in all cases, regardless of whether the messages are correct or not. After forwarding these messages, the receiver sensors will execute the signature verification processes to ensure the correctness of these messages. If the messages are correct, then the sensor will process the messages. Otherwise, the sensor node will drop the messages after the verification process has failed. As a result, the fake messages are spread across the network. Consequently, sensors' energy is consumed by sending, receiving and verifying fake messages. In general, the transmitted messages contain the index of the message (i), the message itself (M) and the broadcast authenticator of this message (BA_i) which is the digital signature in this study. Figure 1 shows the FFP algorithm.

Nevertheless, FFP is usually requested by time-sensitive applications, where the data is transmitted, then verified to avoid any delay of benign messages. However, FFP aids in distributing the malicious messages that deplete the sensors' resources in terms of communication and processing, thus affecting the overall availability of the entire network.

```
Algorithm 1: Forwarding First Protocol Algorithm
Input: msg ( i, M, BAi )
msg = ( i, M, BAi)
forward msg;
Validity = Check_Broadcast_Authenticator (BAi);
if Validity is true Then
    process the message
else                        // Validity is false
    drop msg;
```

Figure 1. The FFP algorithm.

3.2 Authentication First Protocol (AFP)

AFP protocol [41] is another proposed scheme in which the signed messages broadcasted by the BS will be verified first by the sensors before forwarding them to the nearby neighbours. If the messages are correct, the sensor node will forward them, otherwise these messages will be dropped and no forwarding is initiated. Figure 2 shows the AFP algorithm.

```
Algorithm 2: Authentication First Protocol Algorithm
Input: msg ( i, M, BAi )
msg = ( i, M, BAi )
Validity = Check_Broadcast_Authenticator (BAi);
if Validity is true Then
    forward msg;
else                        // Validity is false
    drop msg;
```

Figure 2. The AFP algorithm.

AFP limits the scattering of fake messages to only the first hop neighbours of the attackers; hence, farthest nodes will not be affected. In contrast, the delay caused by the verification process of correct messages cannot be neglected.

3.3 Adaptive Window Protocol (AWP)

Almomani et al. [42] proposed AWP as a compromising solution between FFP and AFP. AWP uses one-way key chain as a weak pre-authenticator to allow the receiver sensor to recognize the fake messages before verifying their authenticity, thus saving the sensor energy from unnecessary verifications. In other words, AWP provides an indicator whether to apply FFP or AFP in each sensor node. Figure 3 illustrates the AWP algorithm.

As demonstrated in Figure 3, the sensor node first checks the weak pre-authenticator; if it is correct, then each sensor node will check its parameter (W). This W represents the maximum number of hops (H) that the broadcast message can forward without being verified (checking the digital signature). If H >= W, then the node will verify the authenticity of the message. After that, if the message is correctly authenticated, then: (1) it will be forwarded after setting the message's hop counter to zero, indicating that the message has just been authenticated and (2) the window size is progressively updated. Otherwise, the message will be dropped and the window size will be reduced.

Window size is updated according to Equation (1) and Equation (2).

$$cw = \alpha cw + (1 - \alpha) AIMD\_W \tag{1}$$

$$w = round(cw) \tag{2}$$

where, *cw* is the current window that is calculated by the AWP, *AIMD_W* is the window size that is computed according to Additive Increase Multiplicative Decrease (AIMD) approach, in which w = ceiling (*w/2*) in case of corrupted message (fake message) and *w* = *w+1* in case of authentic message (correct message); (w) is the final value which is compared to the hop count value. Additionally, α was chosen to be (0.6) with fake messages and (0.5) with authenticated messages based on experiments.

```
Algorithm 3: Adaptive Window Protocol Algorithm
Input: msg ( i, M, BAi, Ki, H)

msg = ( i, M, BAi, Ki, H)
if Hash(Ki ) = Ki-1 Then // weak pre-authenticator
        if H >=W Then   //Authentication first mode
                Validity = Check_Broadcast_Authenticator (BAi);
                if Validity is true Then
                        H =0;  // msg = ( i, M, BAi, Ki, H);
                        forward msg;
                        AIMD_W = cw +1;
                        α = 0.5;
                else   // Validity is false
                        drop msg;
                        AIMD_W =cw /2;
                        α = 0.6;
                end if;
        else // H< W
                H=H+1;
                forward msg;
                Validity = Check_Broadcast_Authenticator (BAi);
                if Validity is true Then
                        AIMD_W = cw +1;
                        α = 0.5;
                else   // Validity is false
                        drop msg;
                        AIMD_W = cw / 2;
                        α = 0.6;
                end if;
        end if;
        Update w :
        cw = α*cw + (1- α)*AIMD_W;
        W = round (cw);
else                          // the Ki is not valid in the chain
        drop msg;
end if;
Return W;
```

Figure 3. The AWP algorithm [42].

Therefore, the *AIMD_W* upon receiving a corrupted message will take a higher ratio than when receiving an authentic message. These ratios could be changed according to the broadcast nature of the network application and its sensitivity. In case of sensitive applications with high security demands, α should be chosen with small values. The maximum window size (max_win) inside each sensor node is determined with respect to the network size or the sensitivity of the network applications. Eventually, the window size will be generated randomly from the interval [1, max_win] for each sensor node.

3.4 Mobility in WSNs

There is a substantial number of mobility models that exist in WSNs. Mobility models are implemented to study the sensor behaviour for different purposes [43]. The dynamic or mobile WSNs (MWSNs) are important due to their major roles in real-world applications. MWSNs are more frequently used than static WSNs [44]-[45]. Additionally, many applications were proposed for mobile base station(s) with a fully static WSN [46]-[47]. Other mobility models could also have a static BS (sink node) and fully dynamic sensor nodes.

Sundus et al. [9] proposed four main mobility models that are considered as the general mobility models; fully static WSN, static sensors with mobile BS network, dynamic sensors with static BS network and fully mobile WSN. These four models were implemented and tested in our study.

3.5 System Architecture

Figure 4 shows the system architecture that will be followed in this research. The purpose is to evaluate the three BA protocols under several mobility circumstances and to measure to what extent this could affect the performance of the authentication services. The main performance measures that were used are the average end-to-end delay of the network and the amount of consumed energy, taking into consideration different network parameters, including attack's intensity, speed and pause time.

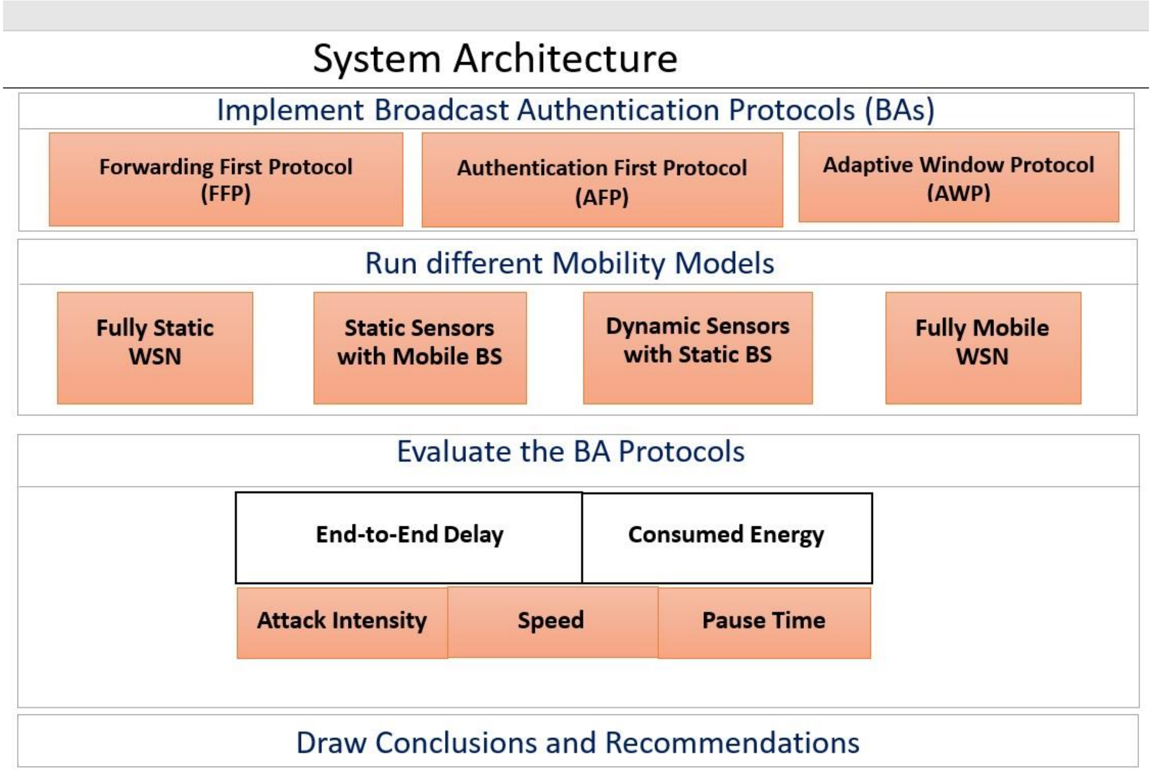

System Architecture
Implement Broadcast Authentication Protocols (BAs)
Forwarding First Protocol (FFP)
Authentication First Protocol (AFP)
Adaptive Window Protocol (AWP)
Run different Mobility Models
Fully Static WSN
Static Sensors with Mobile BS
Dynamic Sensors with Static BS
Fully Mobile WSN
Evaluate the BA Protocols
End-to-End Delay
Consumed Energy
Attack Intensity
Speed
Pause Time
Draw Conclusions and Recommendations

Figure 4. System architecture.

4. Simulation Environment and Evaluation Metrics

Simulation experiments were executed to evaluate the impact of mobility on the performance of BA protocols. This section shows the simulation environment, simulation parameters, evaluation metrics and attacking model.

4.1 Simulation Environment and Parameters

FFP, AFP and AWP were implemented plus evaluated using Qualnet simulator [48]. The detailed simulation parameters that were used to carry out the scenarios are shown in Table 1.

In the proposed simulation environment, the broadcast messages were sent by the BS to the entire sensor network, *via* multiple hops, where some sensor nodes will forward these messages to the neighbours that are far away from the base station. These broadcast messages are either requests or commands; also, the BS signs the message before sending it. After that, each sensor node may perform the message verification to ensure that the message was sent from the BS (trusted message) and not changed or transmitted by the attacker node.

4.2 Evaluation Metrics

The main metrics used to evaluate FFP, AFP and AWP are:

- **Average End-to-end Delay:** This metric analyses the average broadcast delay of the authenticated message in terms of communication and processing time (in seconds) that the message takes until it reaches every node and is processed as well.

- **Average Wasted Energy**: This metric analyses the wasted energy in terms of communication and processing costs (in joules) that is depleted due to injecting the network with fake messages. As a result, the sensor node compels to perform unnecessary operations, such as verifying, sending and receiving these fake messages. Applying security techniques in WSNs to achieve message authenticity requires the sensor node to perform local operations inside each sensor to verify the correctness of the message, dropping energy from the sensor. Moreover, communication cost is the main source for consuming the sensor's energy.

Table 1. Simulation parameter values.

Simulation Parameter	Parameter Value
Number of BS	1 Base Station
Number of Nodes	100 nodes
Simulation Time	1250 seconds
Network Terrain Size	1500 meters X 1500 meters
Node Placement Model	Randomly
Mobility Model	RANDOM-WAYPOINT
Mobility Speed	2.2, 15.5 and 28.8 m/s
Mobility Pause Time	10, 20 and 30 seconds
Transport Layer Protocol	UDP
Digital Signature	ECDSA-160
Window Size (AWP)	6
Transmission Range	250 meters
Packet Size	80 bytes
Packet Sending Interval	30 seconds
Routing Protocol	AODV

- **Attack Intensity:** The purpose of this metric is to examine the behaviour of BA protocols after injecting different intensities of network attackers with different mobile scenarios. Therefore, the attacking model in this research ranges the attack intensity from 0% to 50% of the entire network size.

- **Speed:** Various speeds for mobile nodes were tested using different measurements; meter/second and mile/hour. The approximate mobility speeds are illustrated in Table 2.

Table 2. Approximate mobility speeds.

Scenario	Speed (mph)	Speed (m/s)
Walking	5	2.2
City Driving	35	15.5
Free Way Driving	65	28.8

- **Pause Time:** The pause time of mobile nodes applied in the simulation scenarios was 0, 10, 20 and 30 seconds, respectively, with node speed set to 15.5 meters per second.

4.3 Communication and Processing Costs Analysis

The section provides the analysis for both communication and processing costs in terms of delay and consumed energy.

- ***Communication Delay Analysis***

The communication delay is evaluated in terms of the average time that each message takes to reach the destination, including all possible delays. In other words, the end-to-end delay in mobile WSNs is the time experienced by the message in seconds, which is measured by the generation time of the message at the source node until the message is received by the destination node. In our study, the destination node is every node in the network. The average end-to-end delay for the broadcasting networks [9], [49]-[50] is calculated in Equation (3), where n is the number of messages.

$$\frac{\sum_{1}^{n} message\ recieve\ time - message\ sent\ time}{\sum_{1}^{n} messages\ recieved} \tag{3}$$

- ***Processing Delay Analysis***

The processing cost is evaluated in terms of the number of signature verifications performed on each message during its trip from the BS until reaching the sensor nodes multiplied by the verification time needed for each verification process which is assumed to be 2 seconds in our study [9], [41]-[42]. In other words, the same signature authentication technique was applied by all analyzed protocols to have fair and accurate results. Equation (4) displays the processing delay analysis.

$$\mathit{Processing\ Delay} = 2 * \mathit{number\ of\ verifications} \qquad (4)$$

In more detail, in FFP, each sensor node sends the message before applying the verification process. Thus, after forwarding the message, the verification process is initiated. Hence, the processing delay is not calculated in this protocol. In AFP, each sensor node, before forwarding the message, checks the authenticity of the message first. In case of a fake message, the message will be dropped and the forwarding process will not be initiated. Otherwise, if the message is correct, the message will be forwarded to the next neighbour nodes. AWP is a compromised protocol between FFP and AFP. In this protocol, the most important aspect is the size of the window (which is the number of hops that the message passes without being verified first). Figure 5 shows how delay is changed according to the window size. This study has chosen a window size of 6 to observe the effect of AWP clearly. This window size could be adjusted according to the application sensitivity deployed in WSNs.

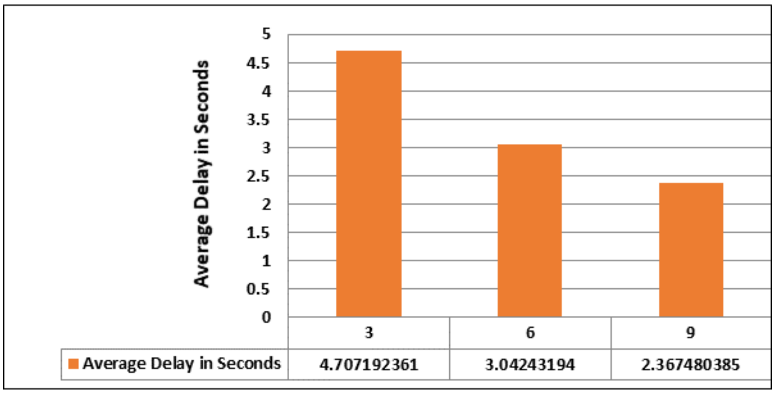


Figure 5. Different window sizes in AWP.

- ***Communication Energy Cost Analysis***

The energy model of the sensor applied in this research is based on the first-order radio model [9], [51]-[55]. Table 3 presents this model.

Table 3. Radio characteristics, first order-radio model [52].

Radio Model (operation)	**Energy Consumption**
Transmitter Electronics ($E_{Tx\,-elec}$) Receiver Electronics ($E_{Rx\,-\,elec}$) ($E_{Tx-elec} = E_{Rx-elec} = E_{elec}$)	50 nJ/bit
Transmit Amplifier (E_{amp})	100 pJ/bit/m^2
Radio Model (operation)	Energy Consumption

The total wasted energy T_x for transmitting a k-bit message is given by Equation (5), where, k is the message size in bits and d is the distance between the sending and the receiving nodes, E_{elec} is the transmitter electronics, E_{amp} is the transmit amplifier. In this study, the average of wasted energy is calculated for retransmitting the fake messages.

$$T_x = E_{elec} * k + E_{amp} * k * d^2 \qquad (5)$$

However, R_x is the total wasted energy for receiving a message which is given in Equation (6), whereby k is the received message size in bytes. In this study, the average wasted energy is calculated for receiving fake messages. Therefore, the overall communication cost will be the total amount of wasted energy for transmitting and receiving fake messages, as shown in Equation (7).

$$Rx = Eelec * k \tag{6}$$

$$\sum Communication\ wasted\ energy = \sum T_x\ (fake\ messages) + \sum R_x\ (fake\ messages) \tag{7}$$

- ***Processing Energy Cost Analysis***

Applying security techniques to any protocol increases the overhead in the network, which consequently increases the depletion of its energy by the verifications executed at each sensor node. Therefore, the processing cost will be the total wasted energy P_x due to fake message verifications, as shown in Equation (8).

$$\sum processing\ wasted\ energy = \sum P_x\ (fake\ messages) \tag{8}$$

Additionally, the security processing cost estimations needed for verifying the messages using different digital signature techniques are measured in milli-Joule [56] and illustrated in Table 4. In this research, we used ECDSA-160 to further examine the variations in the average wasted energy in the three studied protocols.

Table 4. Energy cost estimation for security techniques.

Digital Signature Techniques	**Verification Cost (mJ)**
RSA-1024	14.05
ECDSA-160	53.42

4.4 Attacking Model

If the attacker chooses to affect as many nodes as possible, then the attacker will arrange messages to be transmitted consecutively. Each message created or received by the attacker node will be changed into a fake message and then rebroadcast again.

In FFP, the fake message will be spread throughout the network as there is no review procedure regarding the message before being rebroadcast. Thus, the sensor node as well as the attacker node will rebroadcast the fake messages. Within AFP, sensor nodes close to the attacker would be affected, while sensor nodes far away from the attacker nodes will have limited impact. The fake messages broadcasted from the malicious nodes are dropped by the intermediate nodes. In AWP, sensor nodes close to the attacker will be also affected, but for farther nodes, the impact will be quite limited. Similarly, in AWP, the fake messages are dropped by the intermediate nodes.

5. Simulation Results and Analysis

This section illustrates the experimental simulation results of evaluating the three BA protocols using four different mobility models.

5.1 Average End-to-end Delay

This sub-section presents the average delay in the BA protocols running at different mobile scenarios and attack intensities.

Fully Static Wireless Sensor Network

Figure 6 illustrates the average end-to-end delay in the three protocols against changing the attack intensity. FFP introduces much less amount of average broadcast delay than AFP and AWP; as FFP forwards the message then verifies it. So, there is no consideration regarding message correctness or corruptness. The aim is to forward the message as fast as possible, no matter if the message is trusted or not. On the other hand, AFP has the highest average broadcast delay due to the verification processes that are completed before forwarding the messages again. Therefore, each time the nodes receive a message, the verification process is applied, which delays the message before being rebroadcast again to the next neighbour. AFP ensures that only correct messages will be rebroadcast and fake messages will be dropped. AWP is a compromised protocol between FFP and AFP. When the attack intensity is high, the window size decreases, which consequently increases the number of verifications. But, when the attack intensity is low or no attackers exist in the network, the window will increase to its maximum size, so the number of verifications will be reduced.

A comparison among the three protocols shows that AWP improved the average delay by up to 80.16% compared to AFP. Also, FFP improved the average delay by up to 94.6% and 98.93% compared to AWP and AFP, respectively.

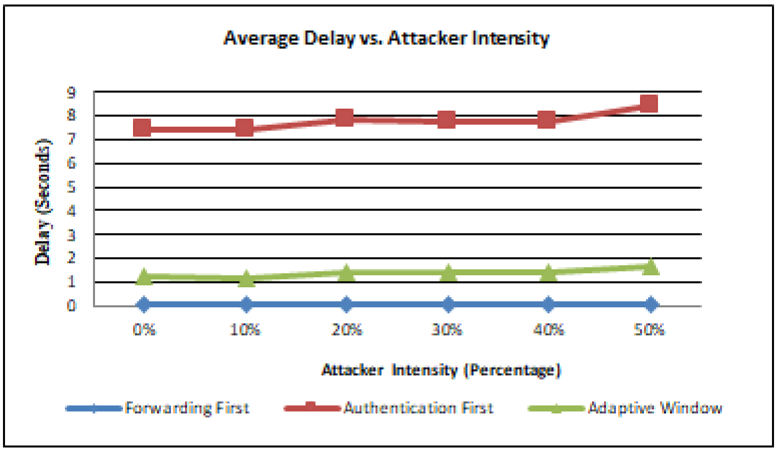


Figure 6. Average delay *vs.* Attacker intensity in static WSN.

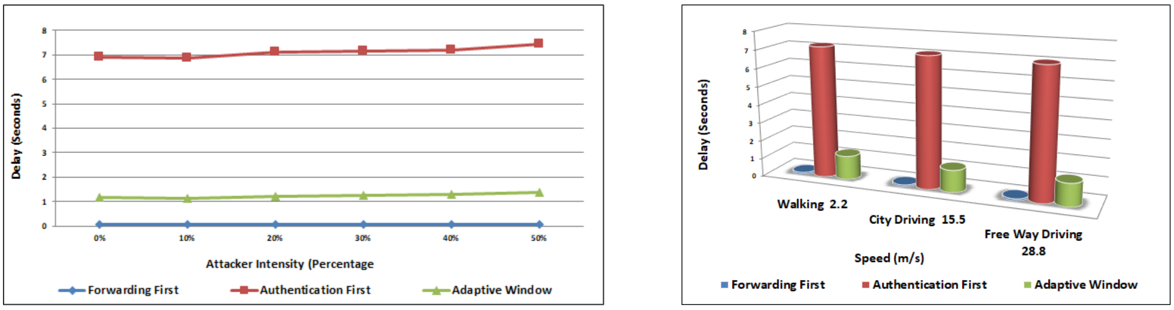
Static Sensors with Mobile Base Station Network

Figure 7(a) shows that AWP improved the average delay by up to 81.32% compared to AFP, whereas FFP improved the average delay by up to 93.62% and 98.81% compared to AWP and AFP, respectively. Figure 7(b) illustrates the average end-to-end delay in the three BA protocols when implementing different speeds. In general, the delay decreased after increasing the sensor speed. Figure 7(c) shows the average end-to-end delay after using different pause times. Overall, as pause time increases, the average delay increases as well.

Dynamic Sensors with Static Base Station Network

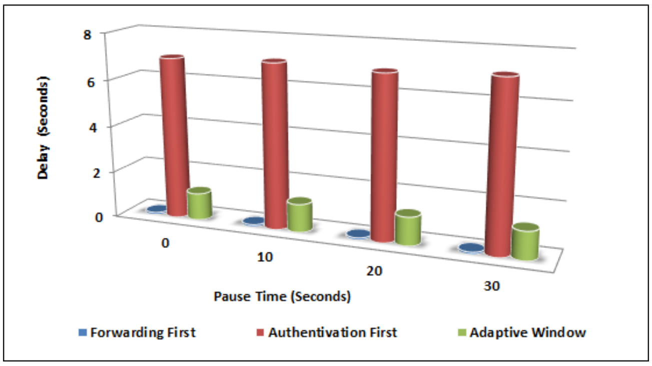
Figure 8(a) illustrates the average end-to-end delay of dynamic sensors with static BS network while changing the attack intensity. As can be observed, AWP improved the average delay by up to 88.24% compared to AFP, whereas FFP improved the average delay by up to 85.63% and 98.31% in comparison with AWP and AFP, respectively.

Figure 8(b) illustrates the average end-to-end delay in the three protocols after applying different speeds. The average delay time in FF protocol decreased by 15.9% due to speed increase. In regard to AFP and the AWP, fewer message loss has occurred, since they allotted time for the verification process before sending out the messages again, consequently giving the sensor node time to locate other viable connections. Further, the average delay increased by up to 63.3% and 73.5% in both AFP and AWP, respectively when there is a boost in speed.



(a) Average delay vs. Attacker intensity
Speed = 15 Pause time = 30

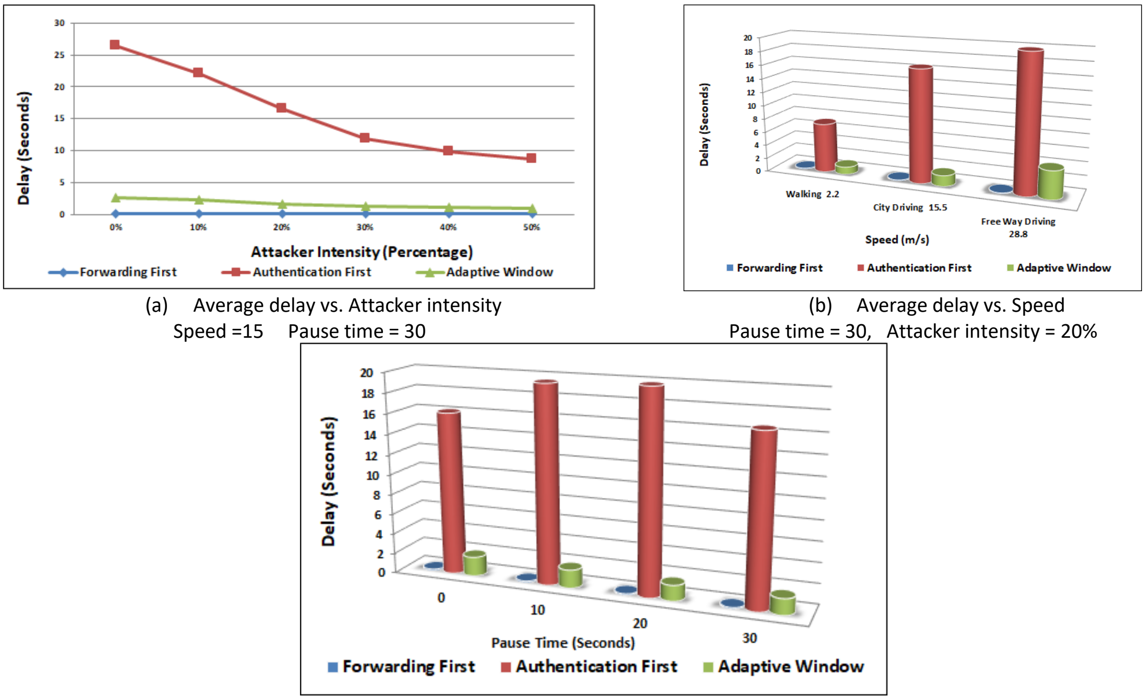
(b) Average delay vs. Speed
Pause time = 30 Attacker intensity = 20%



(c) Average delay vs. Pause time,
Speed = 15 Attacker intensity

Figure 7. Average end-to-end delay in static sensors with mobile BS network.

Figure 8(c) illustrates the decrease in FFP and AWP delay by 7.1% and 16.4%, respectively after increasing the pause time. However, AFP's average delay increased by up to 18.7% until the pause time reached 20, then the average delay decreased.



(a) Average delay vs. Attacker intensity
Speed =15 Pause time = 30

(b) Average delay vs. Speed
Pause time = 30, Attacker intensity = 20%

(c) Average delay vs. Pause time
Speed = 15 Attacker intensity = 20%

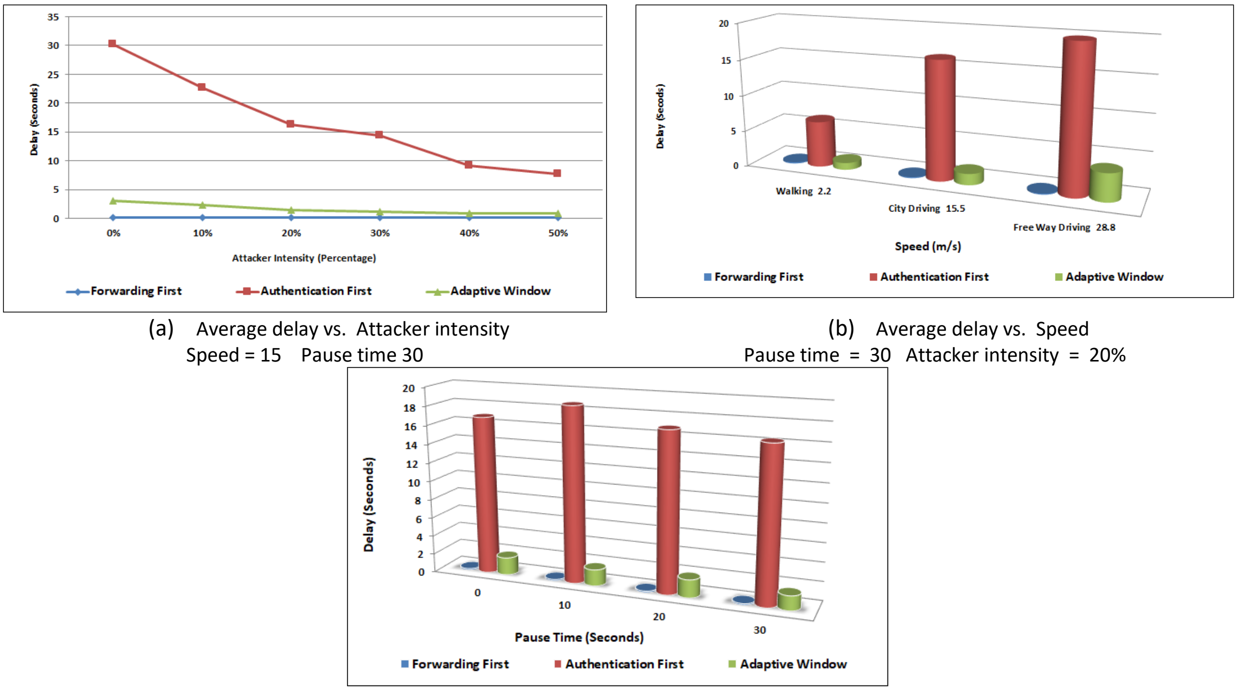
Figure 8. Average end-to-end delay in dynamic sensors with static BS network.

A Fully Mobile Wireless Sensor Network

Figure 9(a) shows the improved delay of AWP over AFP by up to 88.67%, whereas FFP outperformed both AWP and AFP by improving the delay by 83.15% and 98.1%, respectively, considering different attack intensities.

Figure 9(b) illustrates the average delay at different speed values in fully mobile WSN. FFP decreased the average delay by 16.9% while increasing the speed. However, AFP and AWP introduced more delay due to message verification before message forwarding. Thus, AFP and AWP average delay had an increase of 74.38% and 67.68%, respectively during a speed increase.

Figure 9(c) illustrates the average end-to-end delay using different pause times. FFP had an average delay that decreased by 6.8%. However, AFP's delay increased by up to 9.7% until the pause time reached 10, then it was decreased by 12.6%. Moreover, in both AWP and AFP, the delay increased by up to 9% until the pause time reached 20, then it was decreased by 19.1%.



(a) Average delay vs. Attacker intensity
Speed = 15 Pause time 30

(b) Average delay vs. Speed
Pause time = 30 Attacker intensity = 20%

c) Average delay vs. Pause time
Speed = 15 Attacker intensity

Figure 9. Average end-to-end delay in fully mobile WSN.

5.2 Average Wasted Energy

In this sub-section, the BA protocols are evaluated in terms of wasted energy, considering different mobility models, attack intensities, speeds and pause times.

Fully Static Wireless Sensor Network

Figure 10 shows the average wasted energy after forwarding, receiving and verifying fake messages produced by the three protocols under various attack intensities. Comparing the behaviours of FFP, AFP and AWP, it can be observed that the average wasted energy consumed by AWP is small-scale compared to FFP and AFP. The reason is that AWP depends on verifying the weak pre-authenticator each time before forwarding the message. As a result, the AWP discovers fake messages before verifying the broadcast authenticator (digital signature) and stops spreading the fake messages over the network. As can be observed, AFP wasted energy by up to 84.2% less than FFP, whereas AWP wasted energy by up to 65% and 94.4% less than AFP and FFP, respectively.

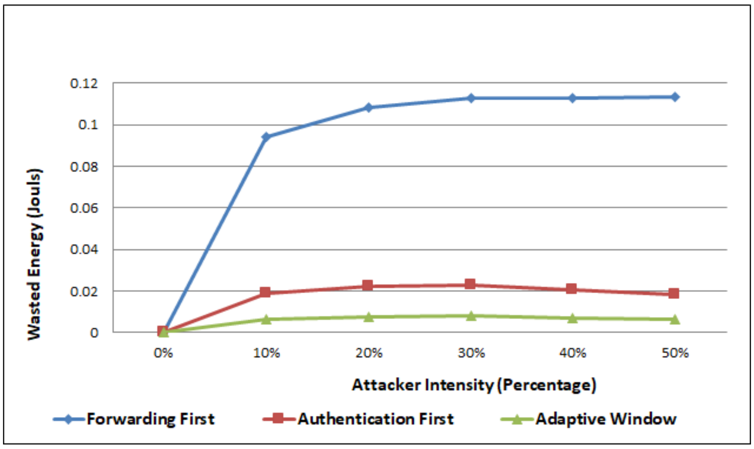


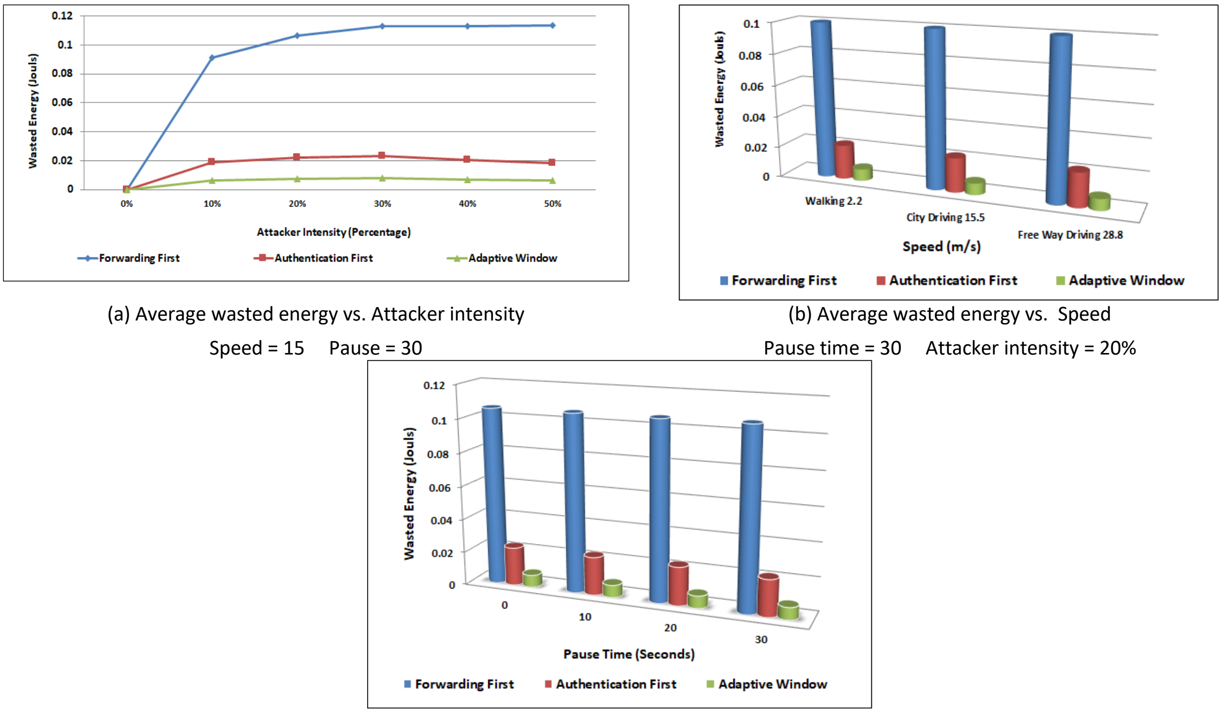
Figure 10. Average wasted energy *vs.* attack intensity in static WSN.

Static Sensors with Mobile Base Station

Figure 11(a) illustrates the average wasted energy in the three protocols while changing the attack intensity. AFP wasted up to 84.04% less energy than FFP. AWP wasted up to 65.5% and 94.5% less energy than AFP and FFP, respectively.

Figure 11(b) shows the average wasted energy produced under different speeds. The results revealed that FFP, AFP and AWP have an energy consumption decrease by 2.02%, 94.5% and 2.8%, respectively during the speed increase. In general, having static sensors with mobile BS network, the average wasted energy slightly decreased while increasing the speed.

Figure 11(c) shows the average wasted energy under different pause times. Comparing the behaviours of FFP, AFP and AWP, it can be noted that the three tested protocols relatively stayed the same, although different pause times were adopted. FFP, AWP and AFP had an energy consumption increase by up to 0.55%, 0.96% and 4.28%, respectively during the increase of pause time. In general, the average wasted energy slightly increases as the pause time increases.



(a) Average wasted energy vs. Attacker intensity

Speed = 15 Pause = 30

(b) Average wasted energy vs. Speed

Pause time = 30 Attacker intensity = 20%

(c) Average wasted energy vs. Pause time

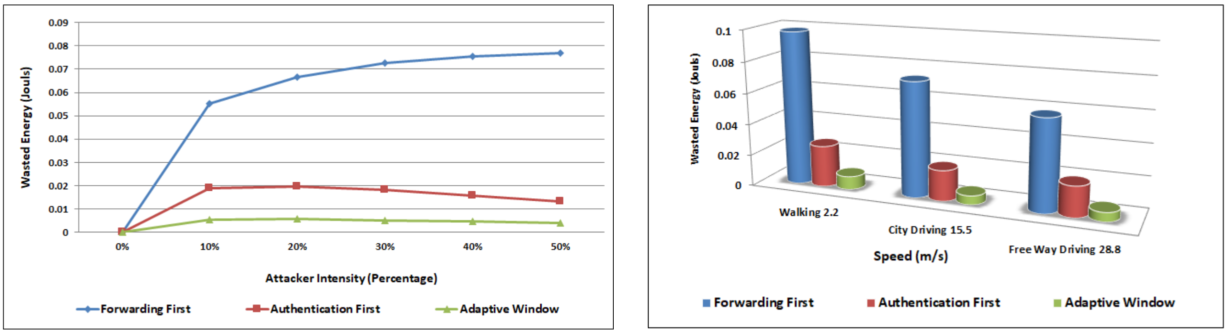
Speed = 15 Attacker intensity = 20%

Figure 11. Average wasted energy in a static sensors and mobile BS network.

Dynamic Sensors with Static Base Station Network

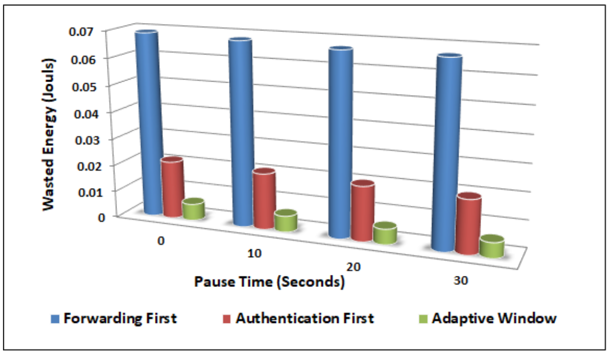
Figure 12(a) illustrates the average wasted energy when changing the attack intensity. AFP wasted energy reached 82.58% less than FFP, whereas AWP wasted energy by up to 70.4% and 94.8% less than AFP and FFP, respectively.

Figure 12(b) displays the average wasted energy spent in processing and communicating fake



(a) Average wasted energy vs. Attacker intensity
Speed = 15 Pause time = 30

(b) Average wasted energy vs. Speed
Pause time = 30 Attacker intensity = 20%



(a) Average wasted energy vs. Pause time
Speed = 15 Attacker intensity = 20%

Figure 12. Average wasted energy in a dynamic sensors with static BS network.

messages produced by the three protocols under different speeds. As can be seen, the energy consumption decreased by 41.02%, 26.53% and 31.57% during speed increase in FFP, AFP and AWP, respectively.

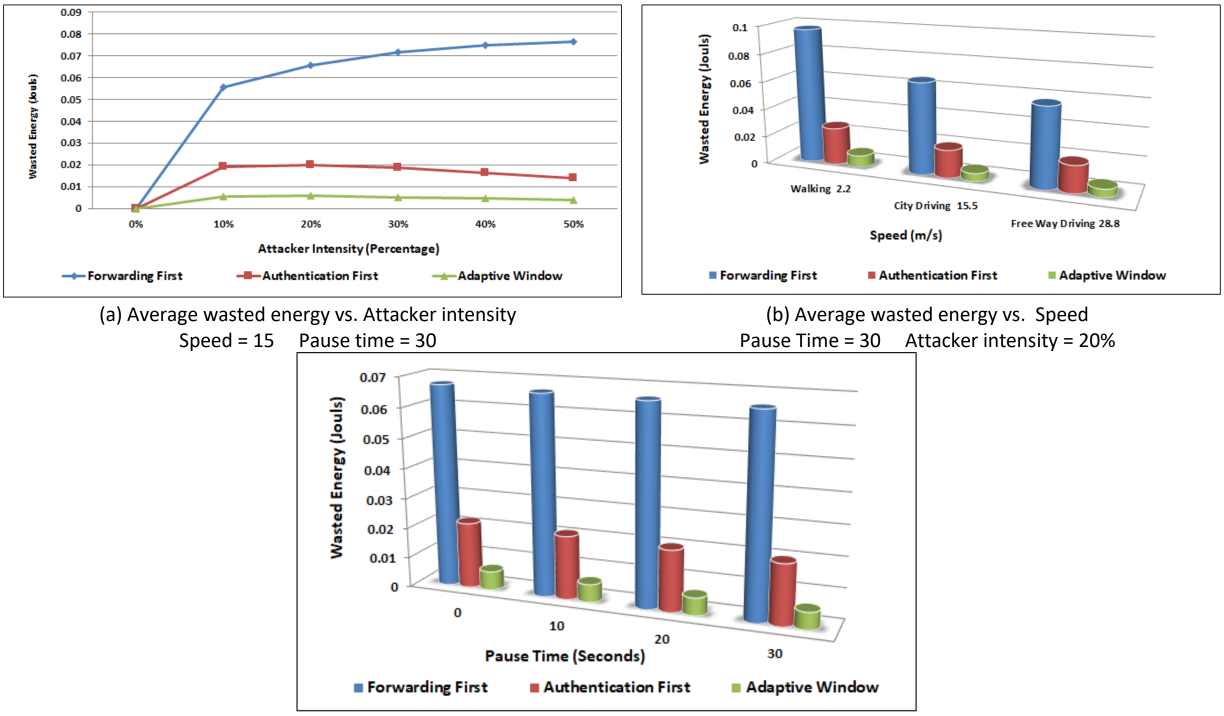
Figure 12(c) illustrates the average wasted energy while applying different pause times. FFP, AFP and AWP consumed energy increased by 3.3%, 8.44% and 8.49%, respectively during long pause times. This is due to receiving additional fake messages from the sensor nodes due to long pause times.

Fully Mobile Wireless Sensor Network

Figure 13(a) illustrates the average wasted energy against attack intensity. AFP wasted energy reached 82.13% less than FFP. Also, AWP wasted energy reached 70.65% and 94.76% less than AFP and FFP, respectively.

Figure 13(b) demonstrates that the average wasted energy decreased by 41.1%, 26.9% and 32.2% in FFP, AFP and AWP, respectively during speed increase.

Figure 13(c) shows the average wasted energy caused by the three protocols under different pause times. FFP, AFP and AWP energy consumption decreased by 2.3%, 8.6% and 7.75%, respectively after increasing the pause time.



(a) Average wasted energy vs. Attacker intensity
Speed = 15 Pause time = 30

(b) Average wasted energy vs. Speed
Pause Time = 30 Attacker intensity = 20%

(c) Average wasted energy vs. Pause time
Speed = 15 Attacker intensity = 20%

Figure 13. Average wasted energy in a fully mobile WSN.

5.3 Summary of Results

Table 5 shows the average delay and the average wasted energy values for FFP, AFP and AWP in all mobility models against 30% attack intensity. This is to facilitate the comparison among the three protocols at a specific attack intensity. The order of protocols in terms of causing less delay in WSN was FFP, AWP and then AFP. In terms of protocols with less consumed energy, the order was AWP, AFP and then FFP, respectively. The differences among these protocols can be observed in Table 5.

Table 5. Average delay and wasted energy for the studied protocols with 30% attack intensity.

The Studied Protocols	Static WSN		Mobile BS with Static WSN		Static BS with Mobile WSN		Fully Mobile WSN	
	Wasted Energy	Delay	Wasted Energy	Delay	Wasted Energy	Delay	Wasted Energy	Delay
FFP	0.11268760	0.09012260	0.11295186	0.08858713	0.07263590	0.14543172	0.07155085	0.14668910
AFP	0.02299114	7.79333621	0.02299347	7.16931291	0.01816784	11.9163161	0.0185301	14.447714
AWP	0.00804712	1.39683641	0.00800553	1.26098829	0.00523681	1.28649027	0.0052342	1.1477775

To provide more comprehensive results, Table 6 summarizes the comparison among FFP, AFP and AWP, where each protocol is compared with itself in case of fully static WSN and when mobility exists. This is to illustrate how the performance of a specific protocol could be affected after introducing mobility. Both FFP and AFP performances were the best in case of mobile BS with static sensors and fully mobile WSN in terms of delay and wasted energy, respectively. On the other hand, AFP performed the best in case of fully mobile WSN and static BS with mobile sensors in terms of delay and wasted energy, respectively. Overall, the best improvements in terms of delay and wasted energy were observed in AWP in comparion with the other two BA protocols.

Table 6. Comparison between each protocol in a fully static WSN with itself using different mobility models.

The Studied Protocols	Mobile BS with Static WSN		Static BS with Mobile WSN		Fully Mobile WSN	
	Wasted Energy	Delay	Wasted Energy	Delay	Wasted Energy	Delay
FFP	0.27% more	1.7% less	31.97% less	38% more	32.3%less	38.5% more
AFP	0.14% more	11.3% less	25.8% more	2.45% more	24.4% less	8.37% less
AWP	1.33% less	16.5% less	37.3% less	13.2% more	36.6% less	47.6% less

6. CONCLUSIONS

Authentication is an important security requirement that needs to be enforced in WSNs. Authentication ensures correct communication between the Base Station (BS) and the sensor nodes. The request or command sent by the BS should be authentic, as it controls the functionality of the WSN and its provided services.

This research examined the effect of mobility on different authentication approaches; the Forwarding First Protocol (FFP), the Authentication First Protocol (AFP) and the Adaptive Window Protocol (AWP) protocols. The performances of FFP, AFP and AWP were experimented against four mobility models: fully static WSN, static sensors with mobile BS, dynamic sensors with static BS network and fully mobile WSN and were measured using different evaluation metrics, including consumed energy, end-to-end delay, speed and pause time.

The simulation results demonstrated that the behaviour of the three BA protocols, which experienced several mobility scenarios, has stayed essentially consistent with differences in the average broadcast delay and the average wasted energy. The average broadcast delay was the best in FFP, but this protocol was the worst in terms of consumed energy. On the other hand, AWP was the best in terms of average wasted energy. Therefore, it can be concluded that AWP was the best protocol in terms of average broadcast delay and average wasted energy, especially when the network is under attack.

For future work, other BA protocols could be tested against mobility models. Also, since the behaviour of protocol has changed in response to mobility, a smart protocol could be designed to flip from one authentication technique to another to maintain efficient authentication services in WSNs.

REFERENCES

[1] B. Mbarek, A. Meddeb, W. Ben Jaballah and M. Mosbah, "A Broadcast Authentication Scheme in IoT Environments," Proc. of the 13th IEEE International Conference of Computer Systems and Applications (AICCSA), Dec. 2016.

[2] F. Wu, X. Li, A. K. Sangaiah, L. Xu, S. Kumari and L. Wu, "A Lightweight and Robust Two-factor Authentication Scheme for Personalized Healthcare Systems Using Wireless Medical Sensor Networks," Future Generation Computer System, vol. 82, pp. 727-737, 2018.

[3] Th. Arampatzis, J. Lygeros and S. Manesis, "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," Proc. of the 13th Mediterranean Conference on Control and Automation, pp. 719-724, 2005.

[4] O. B. Mora, R. Rivera, V. M. Larios, J. R. Beltrán-Ramírez, R. Maciel and A. Ochoa, "A Use Case in Cybersecurity Based in Blockchain to Deal with the Security and Privacy of Citizens and Smart Cities Cyberinfrastructures," IEEE International Smart Cities Conference (ISC2), Sept. 2018.

[5] A. Founoun and A. Hayar, "Evaluation of the concept of the smart city through local regulation and the importance of local initiative", IEEE International Smart Cities Conference (ISC2), USA, Sept. 2018.

[6] P.-A. Mohandas, J. S. A. Dhanaraj and X.-Z. Gao, "Artificial Neural Network based Smart and Energy Efficient Street Lighting System: A Case Study for Residential Area in Hosur," Elsevier, Sustainable Cities and Society, vol. 48, July 2019.

[7] D. J. A. Lewis, "The SMART University: The Transformational Role of Learning Analytics," Information and Learning Science, vol. 119, no. 12, pp. 758-760, 2018.

[8] H. Sharma and G. Kaur, "Optimization and Simulation of Smart Grid Distributed Generation: A Case Study of University Campus," IEEE Smart Energy Grid Engineering (SEGE), Aug. 2016.

[9] K. Sundus and I. Almomani, "Mobility Effect on the Authenticity of Wireless Sensor Networks," Proc. of IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, April 2019.

[10] H. Singh and D. Singh, "Taxonomy of Routing Protocols in Wireless Sensor Networks: A Survey," Proc. of the 2nd International Conference on Contemporary Computing and Informatics (IC3I), pp. 822-830, 2016.

[11] I. Almomani and M. Saadeh, "FEAR: Fuzzy-based Energy Aware Routing Protocol for Wireless Sensor Networks," International Journal of Communications, Networks and System Sciences, vol. 4, no. 6, pp. 403-415, June 2011.

[12] M. Kocakulak and I. Butun, "An Overview of Wireless Sensor Networks towards Internet of Things," Proc. of the 7th IEEE Annual Computing and Communication Workshop and Conference (CCWC), pp. 1-6, 9-11 January 2017.

[13] P. Rawat, K. D. Singh, H. Chaouchi and J. M. Bonnin, "Wireless Sensor Networks: A Survey on Recent Developments and Potential Synergies," Journal of Supercomputing, vol. 68, no. 1, pp. 1–48, April, 2014.

[14] S. K. Gupta and P. Sinha, "Overview of Wireless Sensor Network: A Survey," International Journal of Advanced Research in Computer and Communication Engineering, vol. 3, no. 1, pp. 5201-5207, Jan. 2014.

[15] M. R. Ahmed, X. Huang, D. Sharma and H. Cui, "Wireless Sensor Networks: Characteristics and Architectures," International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, vol. 6, no. 12, pp. 1398-1401, 2012.

[16] W. B. Jaballah, M. Mosbah, H. Youssef and A. Zemmari, "Lightweight Secure Group Communications for Resource Constrained Devices," International Journal of Space- based and Situated Computing, vol. 5, no. 4, pp. 187-200, 2015.

[17] N. K. Mittal, "A Survey on Wireless Sensor Network for Community Intrusion Detection Systems," Proc. of the 3rd IEEE Int'l Conf. on Recent Advances in Information Technology, 2016.

[18] S. Patil, V. Kumar B. P., S. Singh and R. Jamil, "A Survey on Authentication Techniques for Wireless Sensor Networks," International Journal of Applied Engineering Research, vol. 7, no.11, 2012.

[19] B. Mbarek, A. Mddeb, W. Ben Jaballah and M. Mosbah, "An Efficient Broadcast Authentication Scheme in Wireless Sensor Networks," Procedia Computer Science, vol. 109C, pp. 553-559, 2017.

[20] K. Grover and A. Lim, "A Survey of Broadcast Authentication Schemes for Wireless Networks," ELSEVIR *Ad Hoc* Networks, Part A, vol. 24, pp. 288-316, January 2015.

[21] M. Jan, P. Nanda, M. Usman and X. He, "Pawn: A Payload-based Mutual Authentication Scheme for Wireless Sensor Networks," Concurrency and Computation: Practice and Experiance, vol. 29, no. 17, 2016.

[22] V. Khanaa, K. Thooyamani and R. Udayakumar, "A Secure and Efficient Authentication System for Distributed Wireless Sensor Network," World Applied Science Journal (Computer Sceince, Engineering and Its Applications), pp. 304-308, 2014.

[23] I. Almomani and M. Alenezi, "Efficient Denial of Service Attacks Detection in Wireless Sensor Networks," Journal of Information Science and Engineering, vol. 34, no. 4, pp. 977-1000, 2018.

[24] J. P. Walters, Z.-Q. Liang, W.-S. Shi and V. Chaudhary, "Wireless Sensor Network Security: A Survey," Security in Distributed, Grid and Pervasive Computing, p. 367, 2006.

[25] H. Huang, T. Gong, T. Chen, M.-L. Xiong, X.-X. Pan and T. Dai, "An Improved μ TESLA Protocol Based on Queuing Theory and Benaloh-Leichter SSS in WSNs," Journal of Sensors, p. 13, 2016.

[26] M. R. Kumar and C. S. G. Dhas, "An Analysis of Broadcast Authentication and Security Schemes in Wireless Sensor Networks," International Journal of Engineering and Technology (IJET), vol. 5, no. 5, pp. 3992-4001, Nov. 2013.

[27] B. Bezawada, S. Kulkarni and I. Ray, "Independent Key Distribution Protocols for Broadcast Authentication," Symposium on Access Control Models and Technologies (SACMAT 18), pp. 27-38, 13-15 June 2018.

[28] R. Ali, A. K. Pal, S. Kumari, M Karuppiah and M. Conti, "A Secure User Authentication and Key-agreement Scheme Using Wireless Sensor Networks for Agriculture Monitoring," Future Generation Computer Systems, vol. 84, pp. 200-2015, 2018.

[29] S. Challa, A. Kumar Das, V. Odelu, N. Kumar, S. Kumari, M. K. Khan and A. V. Vasilakos, "An Efficient ECC-based Provably Secure Three-factor User Authentication and Key Agreement Protocol for Wireless Healthcare Sensor Networks," Computers and Electrical Engineering, vol. 69, pp. 534-554, July 2018.

[30] C. Ioannou, V. Vassiliou and C. Sergiou, "An Intrusion Detection System for Wireless Sensor Networks," Proc. of the 24th International Conference on Telecommunications (ICT), 3-5 May 2017.

[31] I. Butun, S. D. Morgera and R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," IEEE Communications Surveys & Tutorials, pp. 266 - 282, May 2013.

[32] Krontiris, Intrusion Prevention and Detection in Wireless Sensor Networks, PhD Thesis, Naturwissenschaften der Universit¨at Mannheim, Mannheim, 2008.

[33] O. Karajeh, Securing Wireless Sensor Networks Against Denial of Service Attacks, Thesis for the Master’s Degree of Computer Science, 2010.

[34] K. Han and T. Shon, "Sensor Authentication in Dynamic Wireless Sensor Network Envionments," International Journal of RFID Security and Cryptography (IJRFIDSC), vol. 1, no. 1/2, 2012.

[35] R. Maidhili and G. M. Karthik, "Energy Efficient and Secure Multi-user Broadcast Authentication Scheme in Wireless Sensor Networks," Proc. of IEEE International Conference on Computer Communication and Informatics (ICCCI), Jan. 2018.

[36] D.-H. Lee and I.-Y. Lee, "ECDSA-based Broadcast Authentication Scheme for Smart Home Environments," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 10, no. 4, pp. 81-86, 2018.

[37] H. Bashirpour, S. Bashirpour, S. Shamshirband and A. Chronopoulos, "An Improved Digital Signature Protocol to Multi-user Broadcast Authentication Based on Elliptic Curve Cryptography in Wireless Sensor Networks (WSNs)," Mathematical and Computational Applications, vol. 23, no. 2, pp.17, 2018.

[38] K.-A. Shim, "BASIS: A Practical Multi-user Broadcast Authentication Scheme in Wireless Sensor Networks," IEEE Trans. on Information Forensics and Security, vol. 12, no. 7, pp. 1545-1554, 2017.

[39] C.-Y. Cheng, I.-C. Lin and S.-Y. Huang, "An RSA-like Scheme for Multiuser Broadcast Authentication in Wireless Sensor Networks," International Journal of Distributed Sensor Networks, vol. 11, no. 9, A. ID. 743623, pp. 1-11, 2015.

[40] L. Xu, M. Wen and J. Li, "A Bidirectional Broadcasting Authentication Scheme for Wireless Sensor Networks," Proc. of IEEE Conference on Collaboration and Internet Computing (CIC), pp. 200-204, 2015.

[41] W. Ronghua, D. Wenliang and N. Peng, "Containing Denial-of-Service Attacks in Broadcast Authentication in Sensor Networks," Proceedings of the 8th ACM International Symposium on Mobile *Ad Hoc* Networking and Computing, pp. 71-79, 2007.

[42] I. Almomani, O. Karajeh and L. Abdullah, "Reducing the Vulnerability of Broadcast Authentication against Denial of Service Attacks in Wireless Sensor Networks," The Mediterranean Journal of Computer and Networks, vol. 7, no. 2, 2011.

[43] V. Ramasamy, "Mobile Wireless Sensor Networks: An Overview," Wireless Sensor Networks, [Online], Available: https://www.intechopen.com/books/wireless-sensor-networks-insights-and-innovations/mobile-wireless-sensor-networks-an-overview, October 4th, 2017.

[44] S. M. Mohamed, H. S. Hamza and I. A. Saroit, "Coverage in Mobile Wireless Sensor Networks (M-WSN): A Survey," Computer Communications, vol. 110, pp. 133-150, 15 September 2017.

[45] J. Rezazadeh, M. Moradi and A. S. Ismail, "Mobile Wireless Sensor Networks Overview," International Journal of Computer Communications and Networks (IJCCN), vol. 2, no. 1, February 2012.

[46] N. Ghosh and I. Banerjee, "Application of Mobile Sink in Wireless Sensor Networks," Proc. of the 10th International Conference on Communication Systems & Networks (COMSNETS), 3-7 Jan. 2018.

[47] P. Zhong and F. Ruan, "Application of Mobile Sink in Wireless Sensor Networks Study on the Effect of Sink Moving Trajectory on Wireless Sensor Networks," Proc. of IOP Conference Series: Materials Science and Engineering, vol. 323, 2018.

[48] Scalable Network Technologies, "Qualnet 5.0, Qualnet Network Simulator," [Online], Available: https://www.scalable-networks.com/qualnet-network-simulation.

[49] L. Kumar, "Scalability Performance of AODV, TORA and OLSR with Reference to Variable Network Size," International Journal of Engineering Research and Applications (IJERA), vol. 2, pp. 87-92, 2012.

[50] T. Javed and S. Zafar, "Delay Analysis of Manet Routing Protocols," World Applied Science Journal, vol. 19, no. 5, pp. 615-520., 2012.

[51] J. Banerjee, S. K. Mitra and M. K. Naskar, "Comparative Study of Radio Models for Data Gathering in Wireless Sensor Networks," International Journal of Computer Applications, vol. 27, no. 4, 2011.

[52] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks," Proceeding of the 33rd IEEE Annual Hawaii International Conference on System Science, vol. 2, no. 10, 2000.

[53] H. Aljawawdeh, I. Almomani, "Dynamic load balancing protocol (DLBP) for wireless sensor networks", IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), pp. 1–6, 3-5 Dec. 2013.

[54] H. G. Goh, M. L. Sim and H. T. Ewe, "Energy Efficient Routing for Wireless Sensor Networks with Grid Topology," International Federation for Information Processing (IFIP), pp. 834-843, 2006.

[55] I. Almomani, M. Saadeh, M. AL-Akhras, and H. AL Jawawdeh, “A Tree-Based Power Saving Routing Protocol for Wireless Sensor Networks”, International Journal of Computers and Communications, Vol. 5, no. 2, pp. 84-92, 2011.

[56] I. Almomani and M. Saadeh, "Security Model for Tree-based Routing in Wireless Sensor Networks: Structure and Evaluation," KSII Transactions on Internet and Information Systems (TIIS), vol. 6, no. 4, pp. 1223-1247, 2012.

ملخص البحث:

إن تطبيقات شبكات المجسّات اللاسلكية تتمتع بأهمية كبيرة هذه الأيام، وهي منتشرة في العديد من جوانب حياتنا. وتعدّ بروتوكولات التحقق من أصالة البث حلولاً لضمان أن الأوامر والطلبات التي ترسلها المحطة الأساسية التي تتحكم في الخدمات المقدمة من شبكة المجسات اللاسلكية، هي أوامر وطلبات أصليه. وتعدّ حركية الشبكة أحد التحديات الأساسية التي تواجهها خدمات هذا النوع من الشبكات بشكل عام وبروتوكولات التحقق من الأصالة على وجه الخصوص؛ إذ إن البروتوكولات المتوافرة حاليا للتحقق من الأصالة لم تتفطّن كثيراً الى أثر المحطة الأساسية المتحركة و/أو المجسات المتحركة في سلوك تلك البروتوكولات.

وعليه، تقدم هذه الورقة تحليلاً معمَّقاً لأثر الحركيّة في سلوك بروتوكولات التحقق من أصالة البثّ. وقد تمت دراسة ثلاثة تصاميم مرجعية لبروتوكولات التحقق من أصالة البثّ، هي: التمرير أولاً، والتحقق من الأصالة أولاً، والنافذة التكيفية. وجرى فحص هذه البروتوكولات الثلاثة مقابل أربعة نماذج رئيسية للحركيّة. وكشفت النتائج أنّ بروتوكولات التحقق من أصالة البثّ تصرّفت على نحوٍ مختلف من حيث استهلاك الطاقة والتأخير في الشبكة في وجود الحركية. فعلى سبيل المثال، كان التأخير في بروتوكول النافذة التكيفية قد انخفض بنسبة 47.6% في حالة شبكة المجسات اللاسلكية المتحركة بالكامل، بينما انخفض فقْد الطاقة بنسبة 37.5% في حالة المحطة الأساسية الثابتة والمجسات المتحركة. وعلى الرغم من استخدام تقنية التحقق من الأصالة نفسها للبروتوكولات الثلاثة، فإنّ الحركيّة كانت في حدّ ذاتها سبباً في تحسين الأداء أو تدهوره فيما يتعلق بخدمة التحقق من الأصالة؛ الأمر الذي يؤثر بدوره في أمان شبكات المجسات اللاسلكية والخدمات التي توفرها. فعلى سبيل المثال وفي حالة وجود محطة أساسية متحركة ومجسات ثابتة ، فقد عمل بروتوكول التمرير أولاً على إنقاص التأخير في الشبكة بنسبة وصلت الى 98.81% مقارنة ببروتوكولات التحقق من الأصالة أولاً وبنسبة وصلت الى 93.62% مقارنة ببروتوكول النافذة التكيفية. من جهة أخرى، عمل بروتوكول النافذة التكيفية على توفير طاقة الشبكة بنسبة وصلت الى 94.49% مقارنة ببروتوكول التمرير أولاً وبنسبة وصلت الى 65.5% مقارنة ببروتوكول التحقق من الأصالة أولاً.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 01, March 2020.

BREAST CANCER SEVERITY PREDICATION USING DEEP LEARNING TECHNIQUES

Alaa El-Halees[1] and Mohammed Tafish[2]

(Received: 17-Sep.-2019, Revised: 5-Nov.-2019, Accepted: 30-Nov.-2019)

ABSTRACT

Breast cancer is one of the most common types of cancer most often affecting women. It is a leading cause of cancer death in less developed countries. Thus, it is important to characterize the severity of the disease as soon as possible. In this paper, we applied deep learning methods to determine the severity degree of patients with breast cancer, using real data. The aim of this research is to characterize the severity of the disorder in a shorter time compared to the traditional methods. Deep learning methods are used because of their ability to detect target class more accurately than other machine learning methods, especially in the healthcare domain. In our research, several experiments were conducted using three different deep learning methods, which are: Deep Neural Network (DNN), Recurrent Neural Network (RNN) and Deep Boltzmann Machine (DBM). Then, we compared the performance of these methods with that of the traditional neural network method. We found that the f-measure of using the neural network was 74.52% compared to DNN which was 88.46 %, RNN which was 96.79% and DBM which was 97.28%.

KEYWORDS

Breast cancer severity, Medical data, Deep learning, Deep neural network, Recurrent neural network, Deep Boltzmann machines.

1. INTRODUCTION

Cancer is considered as the second cause of death worldwide and around 70% of cancer cases occur in low- and middle-income countries [1]. Breast cancer is now the foremost cause of cancer-related deaths in women in both of the developed and less developed world. Moreover, the less developed world is suffering from a rising breast cancer disease with a growing number of younger women who are exposed to cancer [2]. In recent times, many health institutions worldwide are working to spread awareness about breast cancer. That is because discovering and treating the disease early reduce number of patients' death.

In breast cancer, tumors appear when cancer makes a mass of tissue in some part of the patient’s body. Various body parts, such as the digestive system, the nervous system or the circulatory system could be affected by these tumors. However, when tumors invade or destroy tissues other parts of the body, they are called metastasized. When a tumor reaches this phase, it becomes harder to treat. Thus, diagnosis time is the most important issue to treat breast cancer. Therefore, it is important to predict the severity of the disease as early as possible before spreading to other parts of the body [3].

For this reason, there are many studies that suggest the use of intelligent methods to detect breast cancer as early as possible. As a result, treatment will be given in a timely manner, which increases the cure rate of the disease.

Traditionally, measuring the severity of breast cancer is carried out manually. For example, doctors manually analyze and interpret data related to patients, disease ….etc. Without using a data analysis system, the analysis will be slow, because experts spend some time to examine the sample. The analysis will also be very subjective, because it depends on the past experience of the expert. Alternatively, using some methods from data mining, such as decision tree, Support Vector Machine, neural networks and deep learning yields faster and more accurate results.

The main objective of this research paper is to use deep learning techniques on the real data of breast cancer patients in order to predict the severity of the disease in these patients. The real data of cancer patients was collected from Gaza strip hospitals.

A. El-Halees and M. Tafish are with Faculty of Information Technology, Islamic University of Gaza, Gaza, Palestine. Emails: [1]alhalees@iugaza.edu.ps and [2]Mtafish14@outlook.com

Breast cancer stage is usually expressed as a number on a scale of 0 through 5, with stage 0 describing non-invasive cancers that remain within their original location, while stage 5 describes invasive cancers, that have spread outside the breast to other parts of the body [4]. In this research, we predict which stage the patient reached. We considered patients in stage 4 and stage 5 as highly severe.

We argued that this study supports the diagnosis of the disease by doctors for predicting a patient's severity condition by applying deep learning methods. Deep learning is a machine learning method that utilizes many levels of artificial neural networks to carry out the process training data. Deep learning has been used in a lot of research on the analysis of medical data, such as works [5]-[12]. We used deep learning because of its ability to detect the target class more accurately than other machine learning methods, especially in the healthcare domain [5].

The rest of the paper is structured as follows: the second section discusses the related work, the third section addresses our material and methods, the fourth section is about experiments and results, the fifth section is a discussion, while the sixth section implies the conclusions and suggestions for future work.

2. RELATED WORK

Due to the large quantity of data in the medical field, many published papers have applied machine language on such data. For example, Danaee, Ghaeini and Hendrix in [6] used a deep learning method called Stacked Denoising Autoencoder (SDAE) to extract functional features from gene expression profiles with high-dimensional data. They used machine learning classification techniques to evaluate the extracted features and found that the new features are very useful in cancer detection. That is because genes, which are highly interactive, could be useful cancer biomarkers for the detection of breast cancer. They concluded that SDAE can be used to extract genes that predict breast cancer and have potential as biomarkers or therapeutic targets.

Fombellida et al. in [7] applied different artificial metaplasticity methods for the diagnosis of breast cancer data. Metaplasticity can be considered as a set of algorithms that have a learning ability based on higher-level properties of biological plasticity. They used neural networks with multilayer perceptrons method at the artificial neuron learning level.

Benzheng et al. in [8] proposed a deep learning method called deep convolutional neural networks in order to classify breast cancer histopathological images. They classified each of the pathological images as one of two breast cancer classes. They found that the model in a prior knowledge that considers class and sub-class labels of breast cancer. That can control the distance of features of different breast cancer histopathological images. They conducted several experiments and the results showed that the model has a classification accuracy of up to 97%, which is a high classification accuracy.

Sekaran, Ramalingam and Mouli in [9] presented a computer-aided diagnosis system to perform automated diagnosis for breast cancer. The system employed deep neural network (DNN) as classifier model and recursive feature elimination (RFE) for feature selection. They used DNN with multiple layers of processing attaining a higher classification rate than SVM. They obtained an accuracy of 98.62%, which is better than other used methods.

Nawaz , Sewissy and Soliman in [10] presented a deep learning approach based on a Convolutional Neural Network (CNN) model for multi-class breast cancer classification. Their method aims to classify breast tumors into benign or malignant, along with predicting the subclass of the tumors, like fibroadenoma, lobular carcinoma …etc. Experimental results using the BreakHis dataset showed that the DenseNet CNN model achieved a high processing performance with 95.4% of accuracy.

Rashed and Abou El Seoud in [11] used a new network architecture inspired by the U-net structure for the early detection of breast cancer using mammograms. The results indicate a high rate of sensitivity and specificity.

Xie et al. in [12] introduced a deep learning method to analyze histopathological images of breast cancer *via* supervised and unsupervised deep convolutional neural networks. They adapted Inception_V3 and Inception_ResNet_V2 architectures to the binary and multi-class issues of breast cancer histopathological image classification by utilizing transfer learning techniques. The experimental results demonstrated that Inception_ResNet_V2 network-based deep transfer learning provides a new means of performing analysis of histopathological images of breast cancer.

From the above research, we can conclude that most of the previous studies concentrated on classifying breast tumors as benign or malignant. But, our work considers the severity of the tumor rather than its classification. Also, the other works applied their methods on histopathological images using computer-aided diagnosis systems, where our work used a dataset, because sometimes such a system is not available. Finally, our work used real system from local medical institutes, whereas other systems mostly used public datasets.

3. Materials and Methods

3.1 Dataset Sources and Description

In our experiments, we used four sources to collect information about breast cancer patients. We collected data from: general hospitals, tissue examination laboratories, radiation centres and death certificates. The data was collected from all hospitals and counselling centres in Gaza city, Palestine, over the period of 4 years from 2011 to 2014. We collected about 721 patients' records. The description of the attributes is depicted in Table 1. Our dataset consists of several attributes and the class is the severity of the disease. The attributes contain general features, such as age, marital status, incidence date and whether the patient is a smoker or not. Also, we have some specific attributes, such as which part of the origin is affected, laterality attribute which describes which side of the origin is affected, morphology attribute which describes the form of carcinogenic cells, the stage diagnosis attribute which describes the originally infected cells and the spread of the disease to neighbouring

Table 1. Attributes description.

Attribute	Description
Marital status	married :1, single: 2, divorced: 3, widowed: 4
Incidence date	Date
Affected part	diagnostic codes
Morphology	NOS =1, Infiltrating duct carcinoma= 2, Juvenile carcinoma of breast= 3, not given=4
Surgery	Given: 1, not given=0
Smoker	yes: 1, No: 0
Laterality	left: 1, right: 2, not paired: 3
Stage	Localized=1, Regional by both direct extension and lymph nodes=2, Regional by direct extension=3, Regional by lymph nodes=4
Radio therapy	Given: 1, not given: 0
Chemical therapy	Given: 1, not given: 0
Immuneotherapy	Given: 1, not given: 0
Hormonal therapy	Given: 1, not given: 0
Clump thickness	Number between 1 and 10
Severity (class)	True = disease dangerous level, False = disease at the beginning

cells. Also, the table describes the radiotherapy attribute which shows whether the patient had radiotherapy or not, the surgery attribute which shows whether the patient had surgery or not, immunotherapy attribute which shows whether the patient had immunotherapy or not, the chemical therapy attribute which gives whether the patient had chemical therapy or not and the hormonal therapy attribute which gives whether the patient had hormonal therapy or not. The last attribute is clump thickness. The class label contains two types of severity, which are high and low.

3.2 Preprocessing

After we integrated data into one dataset, some preprocessing steps have been applied, such as removing repeated patients' records, removing patients with little information and removing some private data columns, such as names and telephone numbers.

After that, data cleaning has been carried out. We used the missing values method in order to replace the missing values, then we set the role (class) of the dataset. In our case, the class attribute is severity, noting that we renamed the class from grade to severity. We used particular features which were selected from the real dataset, so that the classification model used only useful and relevant information.

3.3 Backpropagation Neural Network (BNN)

We used the traditional neural network as the baseline for our experiments. Backpropagation (BP) is a common method for neural networks. Figure 1 gives the architecture of the neural network with backpropagation. BNN is a set of connected nodes called neurons. Neurons are connected by edges. Neurons and edges have weights which are adjusted during the training process. BNN uses loss function to calculates the difference between the predicated output of the neural network and the actual output. Therefore, the goal of using the BNN is to update weights in the network to be as close as possible to the target output, by making the values of the predicated output closer to the values of the network output.

From research, it is found that BNN works only for a small number of hidden layers. From there, we came to the idea of deep learning using more hidden layers [13].

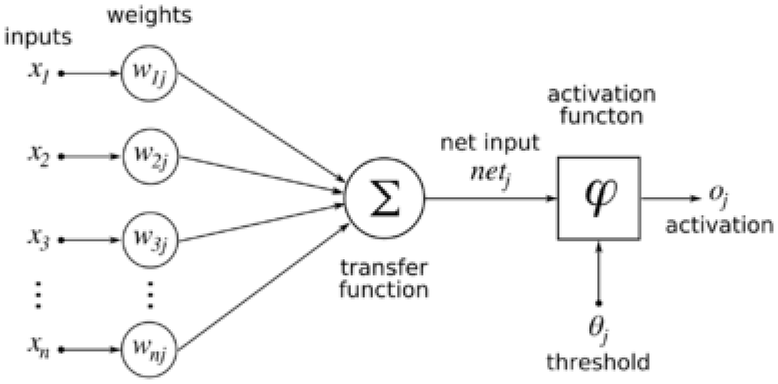


Figure 1. Backpropagation neural network [13].

3.4 Deep Learning Methods

Deep learning is an advanced method that has a collection of algorithms used for building and training neural networks. We used deep learning to improve the performance of traditional neural networks. In deep learning, input data is passed through a set of nonlinear transformation layers to reach the output. We input a set of features and use learning to predict the complex dependencies among these features.

Unlike traditional neural network which builds analysis with data in a linear way, deep learning uses multi-level layers to model high-level abstractions in data, which are composed of multiple nonlinear transformations. In each layer of the deep learning model, nodes train on a separate set of features based on the preceding layer's output [14].

In this paper, we used three common deep learning architectures which are: deep neural network (DNN), recurrent neural network (RNN) and Deep Boltzmann Machine (DBM).

1. ***Deep Neural Network (DNN)***: It is also called multilayer perceptron, which is a multilevel complex neural network. It contains many hidden layers. It uses sophisticated mathematical forms to model data in complex ways.

As seen in Figure 2, DNN is a multilayer perceptron neural network. It contains a number of hidden layers. It is fully connected and each connection has a weight. Each layer contains a number of nodes which is a set of neurons. In DNN, neurons uses nonlinear activation functions except the input neurons [15].

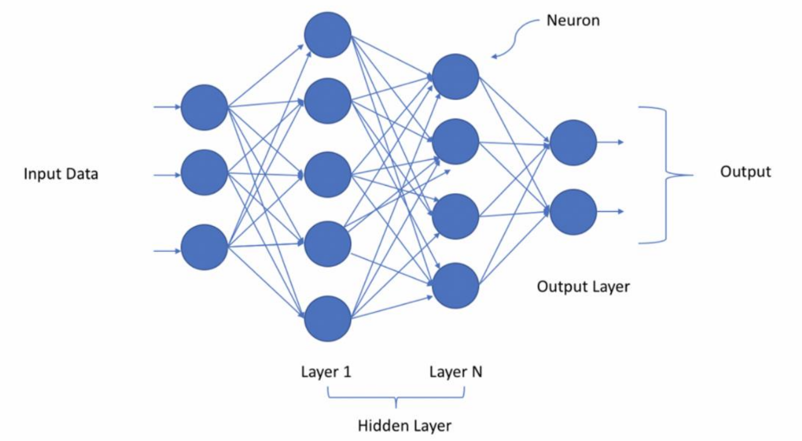


Figure 2. Deep neural network (DNN) [15].

2. ***Recurrent Neural Network (RNN)***: It is another type of deep learning architecture. It connects nodes as a sequence of directed graph. The sequence makes the network have a temporal behavior for a time sequence. As seen in Figure 3, we can use an internal state of RNN nodes as a memory to process a series of inputs. This property differentiates RNN from DNN. Input in RNN not only takes the current input example, but also the examples that appeared previously in time. So, the situation in RNN is unlike in DNN, where inputs and outputs are independent. In RNN, node decisions reached at time step *t-1* affect the decisions reached one moment later at time step *t*. As a result, nodes in RNN have two sources of input; the present and the recent past [16].

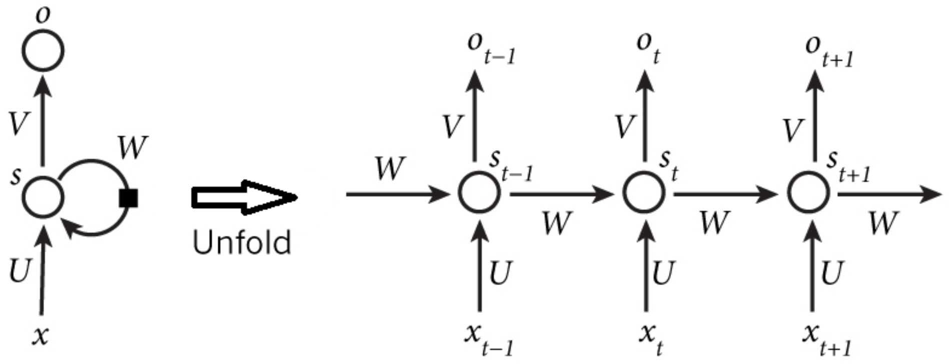


Figure 3. Recurrent neural network (RNN) [17].

3. ***Deep Boltzmann Machine (DBM):*** It is deep multilayer architecture based on Restricted Boltzmann Machine (RBM). The Boltzmann machine is a network of symmetrically coupled stochastic binary units. In DBM, we have a set of units *v ∈{0, 1}* as well as a set of hidden units *h ∈ {0, 1}*. As seen in Figure 4, DBM can be considered as an undirected probabilistic graphical model which contains a layer of observable features and a multilayer of hidden features. We often use DBM as a building block for constructing DNN and deep generative models which have recently gained popularity to learn complex and large probabilistic models [18].

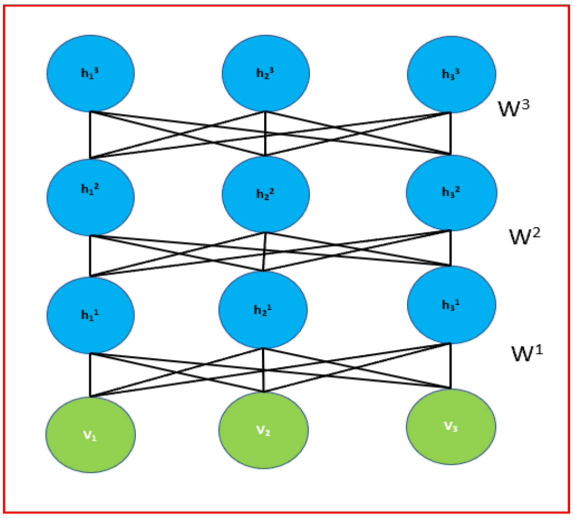


Figure 4: Deep Boltzmann machine (DBM) [18].

3.5 Evaluation

To evaluate our experiment, we used the most common metrics in this area, which are: accuracy, precision, recall and f-measure. In our experiment, we used 10-cross-validation testing. Then, we computed accuracy, which measures the percentage of the test sets that the classifier has labeled correctly. Also, we computed precision, which is the percentage of positive identifications that are actually correct. Then, we computed recall, which is the percentage of actual positives that are correctly identified. Finally, we computed the f-measure, which is a combined metric that takes both precision and recall into consideration.

4. Experimental Results

To experiment the proposed methods, we performed a two-class classification task to discriminate breast cancer severity as severe or not. In our experiments, we used a real breast cancer dataset which contains 721 data points of each class. We used four neural network methods which are:

4.1 Backpropagation Neural Network

To experiment this method, we modelled the network with a sigmoid activation function on all neurons of the network. We trained the model for 500 epochs, with a learning rate of 0.5. Table 2 gives the results of applying NNB. The accuracy was 75.37% and f-measure was 74.54%. This is a particularly weak result in the medical field which needs a highly confident result.

Table 2. Neural network backpropagation (NNB) performance.

Accuracy	75.37%
Recall	70.88%
Precision	78.55%
f-measure	74.52%

4.2 Deep Neural Network (DNN)

To test our data set on DNN, we used three hidden layers with 10, 20 and 10 nodes, as well as 5000 epochs. The evaluation function used was sigmoid function with a dropout of 0.2. Also, we used

Table 3. Deep neural network (DNN) performance.

Accuracy	87.83%
Recall	88.09%
Precision	88.84
f-measure	88.46%

Adam optimizer. As seen in Table 3, using the DNN with this configuration, we obtained an accuracy of 87.8% and an f-measure of 88.46%. These results are better than those obtained using BNN but are still not sufficient for the medical domain.

4.3 Recurrent Neural Network (RNN)

We conducted the third set of experiments using RNN. We used two hidden layers with 5 units for each hidden layer. We, also, used 1000 as maximum iteration to learn. The learning function was standard backpropagation for partial recurrent networks. We set the activation function of the output units to logistic function. We got an accuracy of 96.56% and an f-measure of 96.79%. These results are better than those obtained using BNN and DNN, as shown in Table 4.

Table 4. Recurrent neural network (RNN) performance.

Accuracy	96.56%
Recall	96.96%
Precision	96.63%
f-measure	96.79%

4.4 Deep Boltzmann Machines (DBM)

The fourth method that we used was DBM. In this experiment, we used three hidden layers and 10 nodes for each hidden layer. The learning rate was 0.8 and we used a sigmoid function as the activation function. The number of epochs was 3 and the batch size was 100. Table 5 gives the results of using this method with an accuracy of 97.52% and f-measure of 97.28%.

Table 5. Deep Boltzmann machines (DBM) performance.

Accuracy	97.52%
Recall	97.03%
Precision	97.54%
f-measure	97.28%

5. Discussion

As seen in Figure 5, the worst result came from using Neural Network Backpropagation with an accuracy of 75.37%. Compared to deep learning methods, BNN is too low, mainly because deep learning methods have many nonlinear transformation layers which make them able to detect the complexity of data in complex domains such as the breast cancer domain.

On the other hand, DBM is the best method that we can use to predict severity from breast cancer medical data. That is because of its ability to learn complex and large probabilistic models.

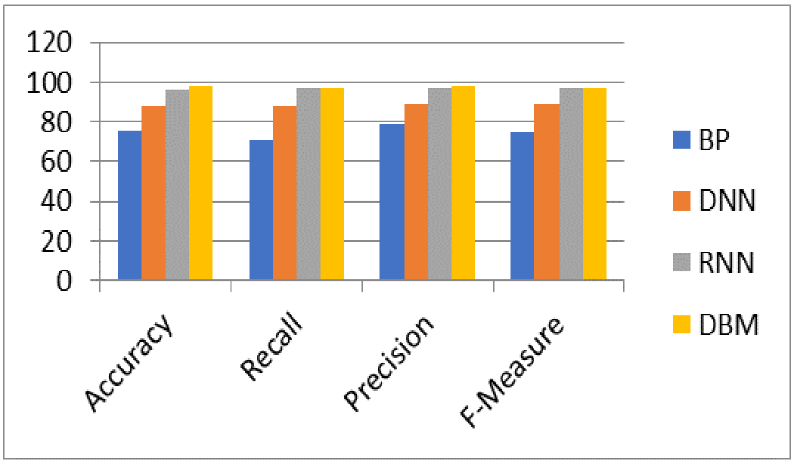


Figure 5. Comparing the performances of the four neural networks methods.

6. CONCLUSIONS

In this paper, different neural network models have been investigated and applied to find the model that best predicts breast cancer severity. The main concern of this paper is to classify patterns of breast cancer dataset into two categories: low severity and high severity of breast cancer grade. We applied four models of neural networks and deep learning: Backpropagation Neural Network (BNN) as baseline, Deep Neural Network (DNN), Recurrent Neural Network (RNN) and Deep Boltzmann Machine (DBM). We found that in general, performance of deep learning methods is much better than that of the traditional neural network. Also, we found that Deep Boltzmann Machine can produce the best results with an accuracy of 97.52% and an f-measure of 97.28%.

In the future, more patients' data will be collected to make a bigger training dataset for further testing and evaluation in order to increase the severity detection hit rate and improve model accuracy. Also, we may integrate histopathological images with our dataset as input to the system for more generalizable results.

ACKNOWLEDGEMENT

This research was supported by Qatar Charity under Ibhath project for research grants, which is funded by the Cooperation Council for the Arab States of the Gulf through the Islamic Development Bank.

REFERENCES

[1] World Health Organization (WHO), "Cancer," [Online], Available: http://www.who.int/mediacentre/factsheets/fs297/en/, [Accessed on 12-9-2018].

[2] R. Oskouei, N. Kor and S. Maleki. "Data Mining and Medical World: Breast Cancer's Diagnosis, Treatment, Prognosis and Challenges," American Journal of Cancer Research, vol. 7, no. 3, pp. 610-627, Mar. 2017.

[3] Cleveland Clinic, "Breast Cancer," [Online], Available: https://my.clevelandclinic.org/health/diseases/3986-breast-cancer, [Accessed on 20-8-2018].

[4] Breastcancer.org, "Breast Cancer Stages," [Online], Available: https://www.breastcancer.org/symptoms/diagnosis/staging, [Accessed on 26-10-2018].

[5] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo and G.-Z. Yang, "Deep Learning for Health Informatics," IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 1, pp. 4–21, 2017.

[6] P. Danaee, R. Ghaeini and D. Hendrix. "A Deep Learning Approach for Cancer Detection and Relevant Gene Identification," Pacific Symposium on Biocomputing, vol. 2017, no. 22, pp. 219-229, 2017.

[7] J. Fombellida, S. Torres-Alegre and J. A. Piñuela. "Metaplasticity for Deep Learning: Application to WBCD Breast Cancer Database Classification," J. M. Ferrández Vicente, J. R. Álvarez-Sánchez, F. de la Paz López, F. J. Toledo-Moreo, H. Adeli (Eds.), "Bioinspired Computation in Artificial Systems," (IWINAC 2015), Lecture Notes in Computer Science, vol. 9108, Springer, Cham, 2015.

[8] W. Benzheng, H. Zhongyi, H. Xueying and Y. Y. Yin, "Deep Learning Model-based Breast Cancer Histopathological Image Classification," Proc. of the 2nd IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, pp. 348-353, 2017.

[9] K. Sekaran, S. Ramalingam and C. Mouli, "Breast Cancer Classification Using Deep Neural Networks," S. Margret Anouncia and U. Wiil (Eds.), Knowledge Computing and Its Applications, Springer, Singapore, February 2018.

[10] M. Nawaz, A. Sewissy and T. Soliman, "Multi-class Breast Cancer Classification Using Deep Learning Convolutional Neural Network," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 9, no. 6, 2018.

[11] E. Rashed and A. Abou El Seoud, "Deep Learning Approach for Breast Cancer Diagnosis," Proceedings of the 8th International Conference on Software and Information Engineering, Cairo, Egypt, pp. 243-247, 09 – 12 April 2019.

[12] J. Xie, R. Liu, J. Luttrell and C. Zhang, "Deep Learning-based Analysis of Histopathological Images of Breast Cancer," Frontiers in Genetics, vol. 10, no. 80, 19 Feb. 2019.

[13] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," Neural Networks, vol. 61, pp. 85–117, 2016.

[14] M. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

[15] R. Collobert and J. Weston. "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," Proceedings of the 25th International Conference on Machine Learning (ICML '08), ACM, New York, NY, USA, pp. 160-167, 2008.

[16] T. Mikolov, M. Karafiát, L. Burget, J. Černocký and S. Khudanpur. "Recurrent Neural Network-based Language Model," Proc. of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH-2010), pp. 1045-1048, 2010.

[17] D. Guota, "Fundamentals of Deep Learning–Introduction to Recurrent Neural Networks," [Online], Available: https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/, [Accessed on 20-8-2018].

[18] R. Salakhutdinov and H. Larochelle, "Efficient Learning of Deep Boltzmann Machines," Journal of Machine Learning Research — Proceedings Track, vol. 2010, no. 9, pp. 693–700, 2010.

ملخص البحث:

يُعدّ سرطان الثدي واحداً من أكثر أنواع مرض السرطان شيوعاً، علماً بأنه يصيب النساء في معظم الحالات. وسرطان الثدي يأتي في مقدمة الأسباب التي تؤدي الى حالات الوفاة بمرض السرطان في الدول الأقل نمواً. لذا، فإن من المهم الوقوف على شدة الإصابة بهذا المرض في أسرع وقت ممكن.

في هذه الورقة، طبقنا عدداً من طرق التعلُّم العميق لتحديد شدة إصابة المرضى بهذا المرض باستخدام بيانات حقيقية. ويهدف هذا البحث الى تحديد شدة الإصابة بسرطان الثدي في وقت أقصر مقارنة بالطرق التقليدية. وتستخدم طرق التعلُّم العميق نظراً لما تتمتع به من قدرة على الكشف عن الصنف المستهدف على نحو أدق مقارنة بسواها من طرق تعلُّم الآلة، وبخاصة في ميدان الرعاية الصحية.

يتضمن هذا البحث إجراء عدة تجارب باستخدام ثلاثٍ من طرق التعلُّم العميق المختلفة، هي: الشبكة العصبية العميقة، والشبكة العصبية المتكررة، وآلة بولتزمان العميقة. بعد ذلك، تمت مقارنة أداء هذه الطرق بأداء طريقة الشبكة العصبية التقليدية. وقد تبين أنّ مقياس (ف) لاستخدام الشبكة العصبية بلغ 74.5%، مقارنة بِ 88.46% لطريقة الشبكة العصبية العميقة، و 96.79% لطريقة الشبكة العصبية المتكررة، و 97.28% لطريقة آلة بولتزمان العميقة.

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) مجلة علمية عالمية متخصصة محكمة تنشر الأوراق البحثية الأصيلة عالية المستوى في جميع الجوانب والتقنيات المتعلقة بمجالات تكنولوجيا وهندسة الحاسوب والاتصالات وتكنولوجيا المعلومات.

تحتضن جامعة الأميرة سمية للتكنولوجيا (PSUT) المجلة الأردنية للحاسوب وتكنولوجيا المعلومات، وهي تصدر بدعم من صندوق دعم البحث العلمي في الأردن. وللباحثين الحق في قراءة كامل نصوص الأوراق البحثية المنشورة في المجلة وطباعتها وتوزيعها والبحث عنها وتنزيلها وتصويرها والوصول إليها. وتسمح المجلة بالنسخ من الأوراق المنشورة، لكن مع الإشارة إلى المصدر.

الأهداف والمجال

تهدف المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) إلى نشر آخر التطورات في شكل أوراق بحثية أصيلة وبحوث مراجعة في جميع المجالات المتعلقة بالاتصالات وهندسة الحاسوب وتكنولوجيا المعلومات وجعلها متاحة للباحثين في شتى أرجاء العالم. وتركز المجلة على موضوعات تشمل على سبيل المثال لا الحصر: هندسة الحاسوب وشبكات الاتصالات وعلوم الحاسوب ونظم المعلومات وتكنولوجيا المعلومات وتطبيقاتها.

الفهرسة

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات مفهرسة في كل من:

فريق دعم هيئة التحرير

المحرر اللغوي	ادخال البيانات وسكرتير هيئة التحرير
حيدر المومني	إياد الكوز

عنوان المجلة

الموقع الإلكتروني: www.jjcit.org
البريد الإلكتروني: jjcit@psut.edu.jo
العنوان: جامعة الاميرة سمية للتكنولوجيا، شارع خليل الساكت، الجبيهة، عمان، الأردن.
صندوق بريد: 1438 عمان 11941 الأردن
هاتف: +962-6-5359949
فاكس: +962-6-7295534

Princess Sumaya University for Technology
Since 1991 تأسست
PSUT
جامعة الأميرة سمية للتكنولوجيا

صندوق دعم البحث العلمي
Scientific Research Support Fund

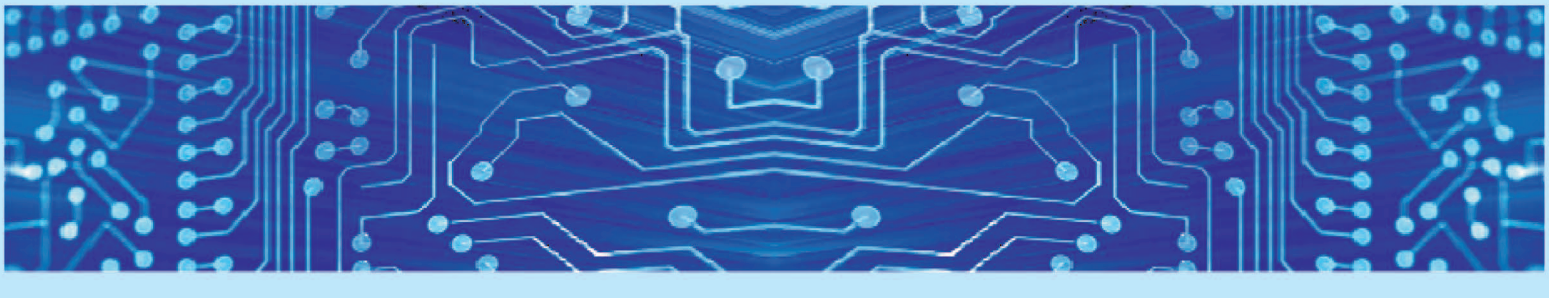
المجلة الأردنية للحاسوب و تكنولوجيا المعلومات

آذار ٢٠٢٠ المجلد ٦ العدد ١
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

الصفحات عنوان البحث

www.jjcit.org jjcit@psut.edu.jo



مجلة علمية عالمية متخصصة محكمة

تصدر بدعم من صندوق دعم البحث العلمي