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

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

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

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]:

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

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

$$\sum_i \left[\left(\frac{(-w_i - x).(w_i - x)^T}{(2\alpha^2)}\right)\right] \quad (2)$$

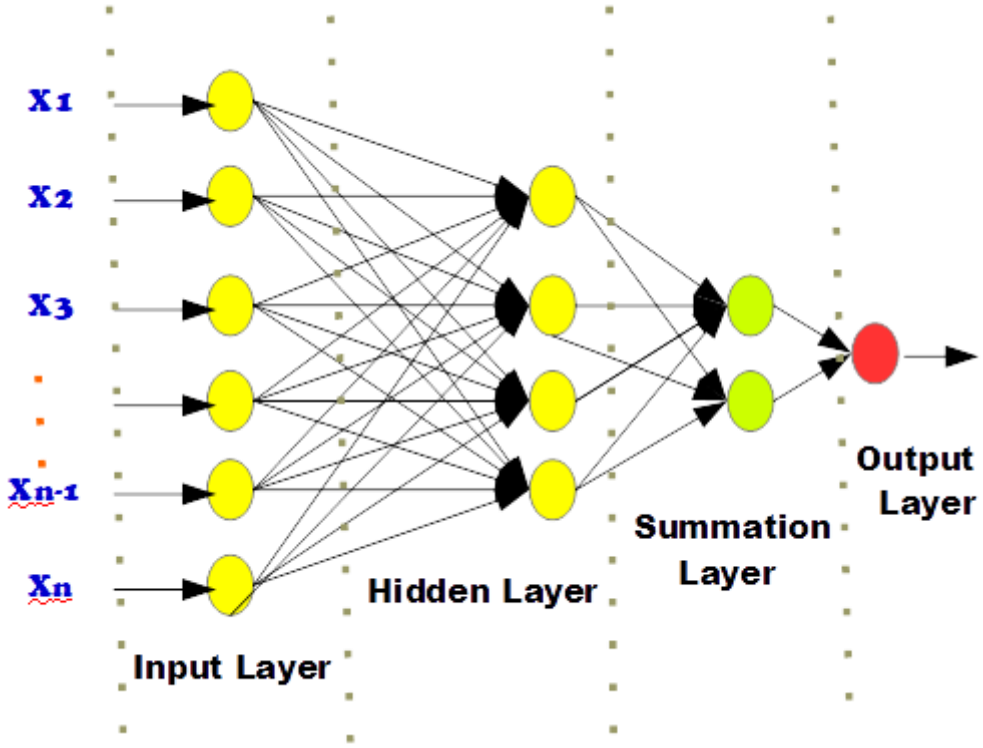


Figure 1. Structure of probabilistic neural network.

- 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:

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

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:

$$C = -\frac{h_s l_s}{h_r l_r} \cdot \frac{n_r}{n_s} \quad (4)$$

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].

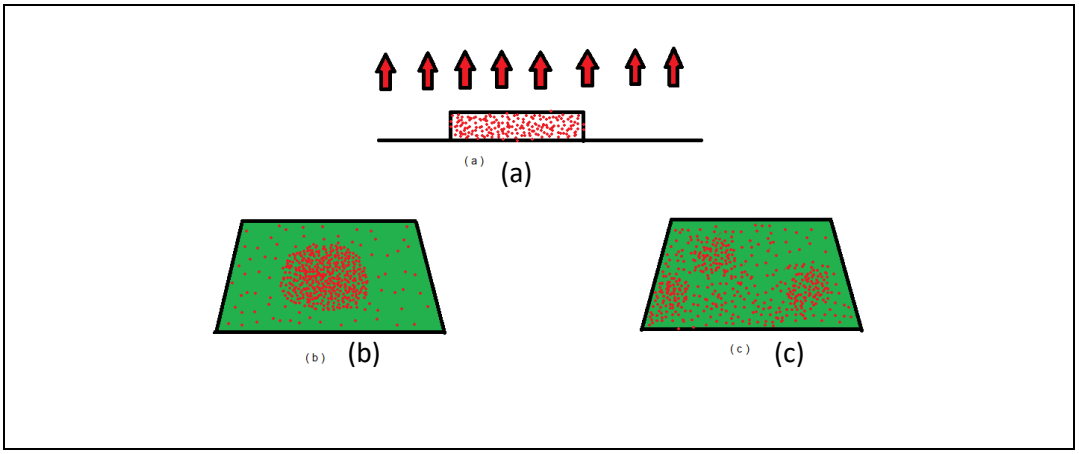


Figure 2. (a) View of initial system; (b) Snapshot of water on a substrate with low wettability; (c) Snapshot of water on a substrate with high wettability [1].

- 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]:

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

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

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

- Generating a random permutation-based step-size matrix

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

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

- 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:

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

- **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:

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

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

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

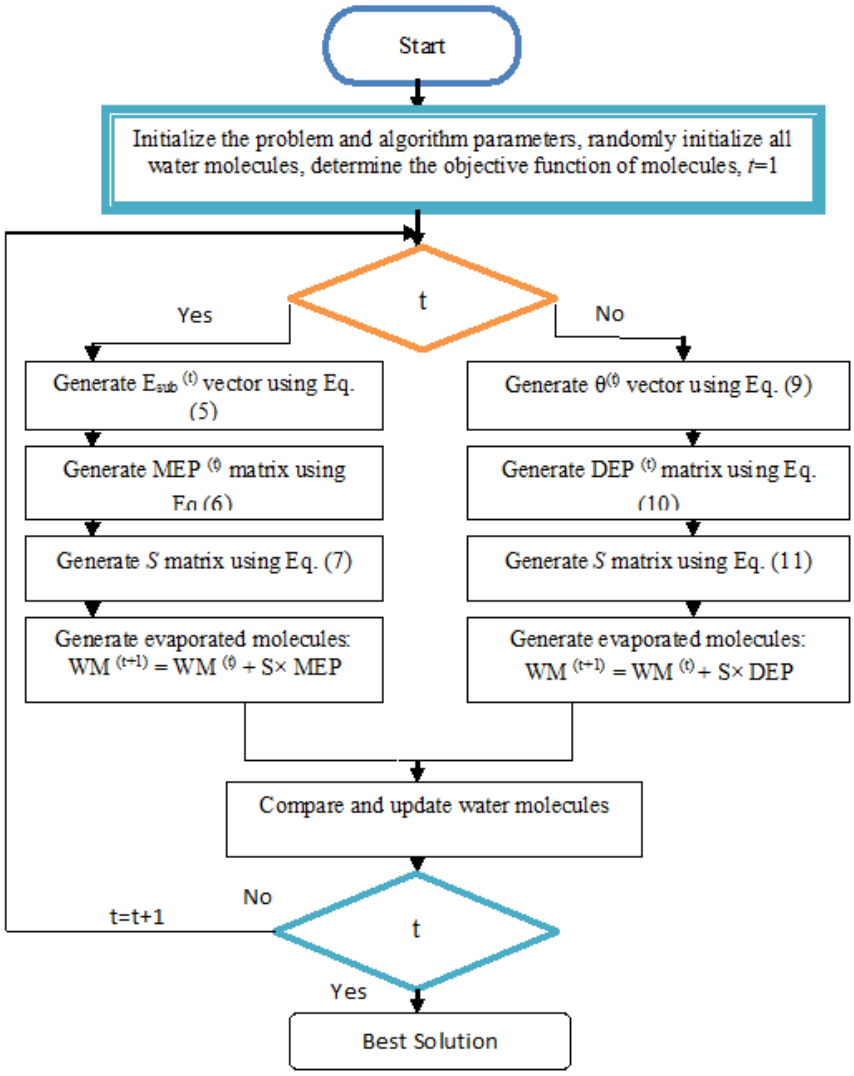


Figure 3. Water Evaporation Algorithm (WEA).

- Generating a random permutation-based step-size matrix

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

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

- 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]:

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

- 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.

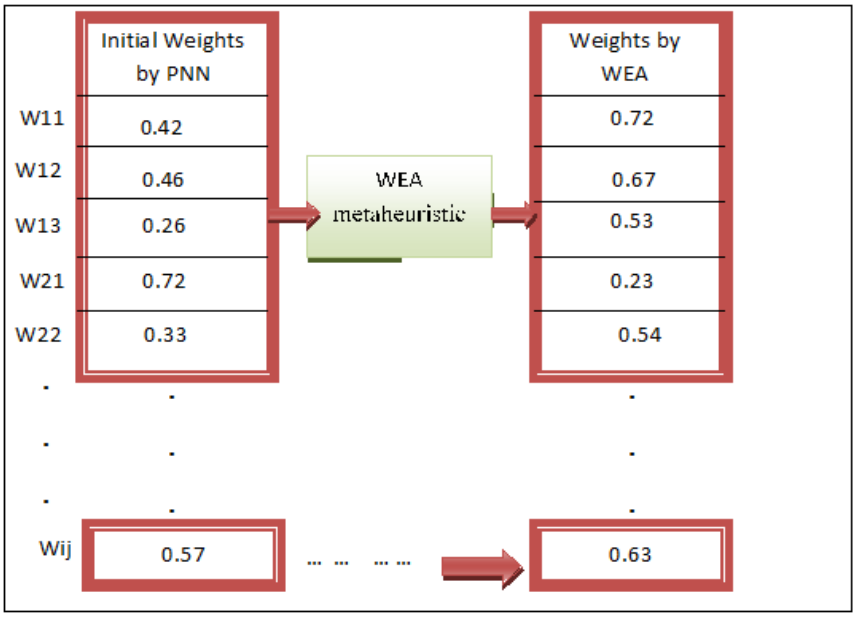


Figure 4. Representation of initial weights.

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.

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

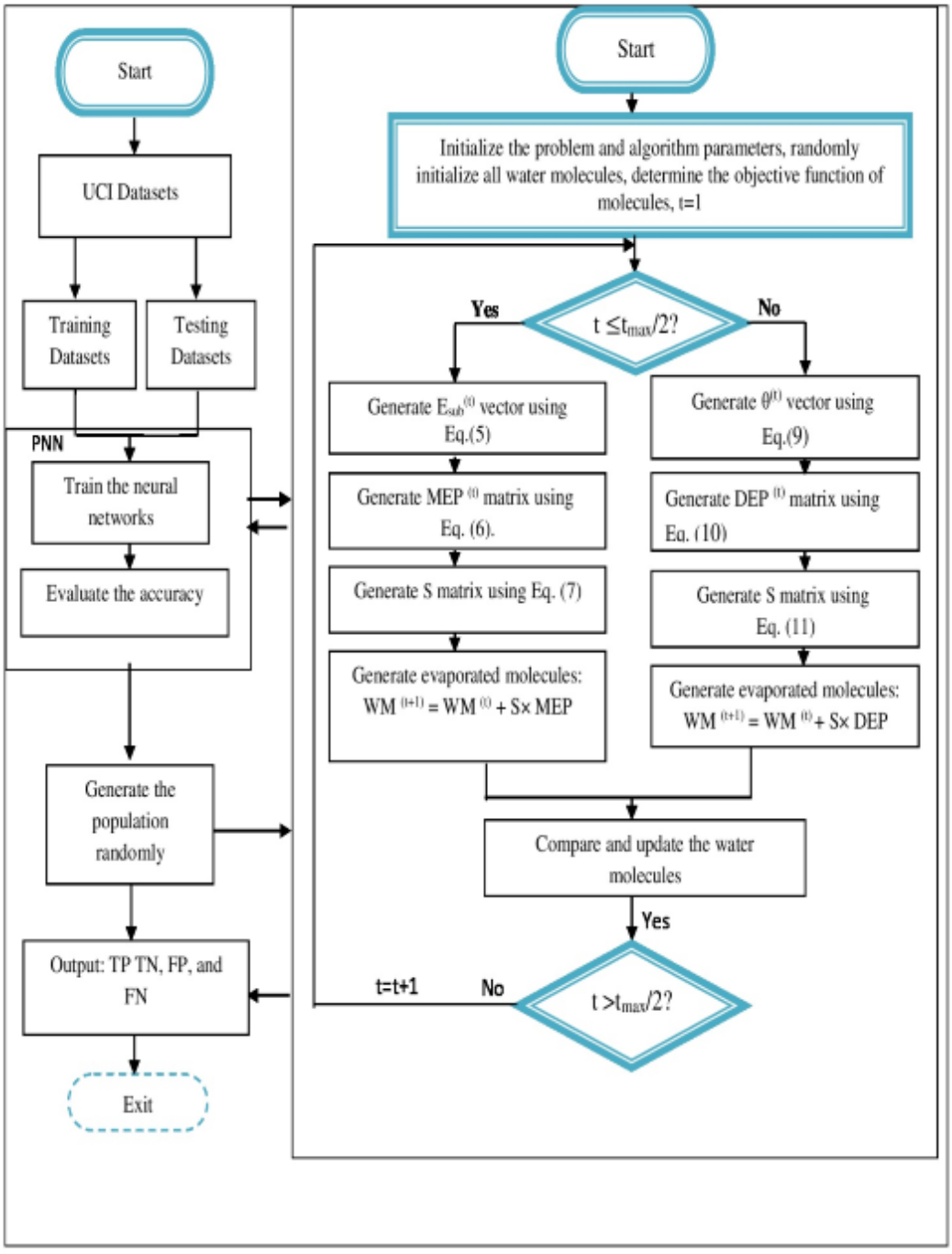


Figure 5. Flowchart of WEA-PNN technique.

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% ($\alpha = 0.05$) 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.

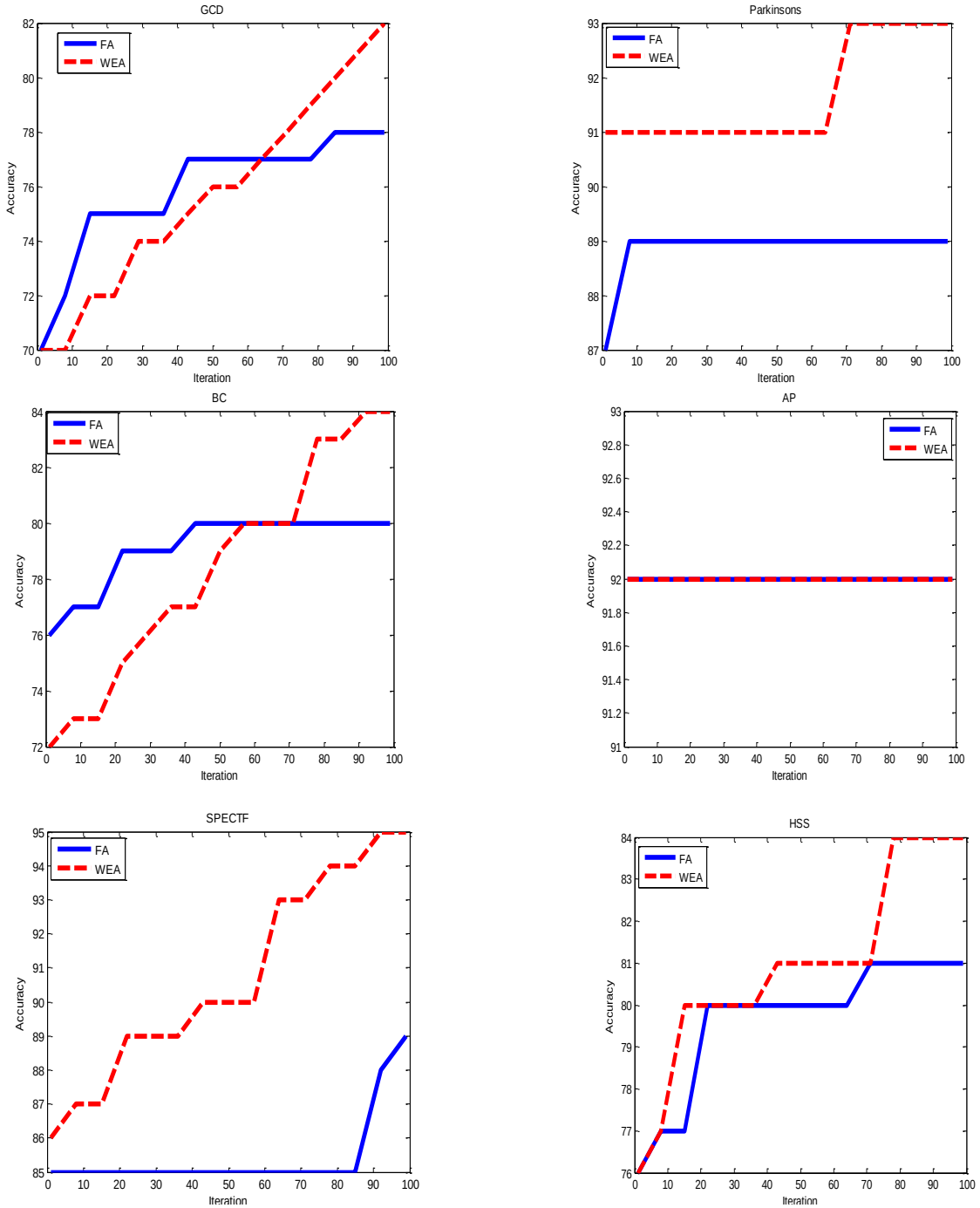
Dataset		**Mean**	**Std. deviation**	**Std. error mean**	**P-value**
PID	WEA	83.1028	0.63377	0.11570	0.00
	FA	73.4895	1.28560	0.23472	
HSS	WEA	83.3766	0.62887	0.11482	0.00
	FA	81.8179	1.02322	0.18681	
AP	WEA	92.5926	0.00000	0.00000	0.00
	FA	92.5926	0.00012	0.00002	
BC	WEA	82.9167	1.59613	0.29141	0.00
	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).



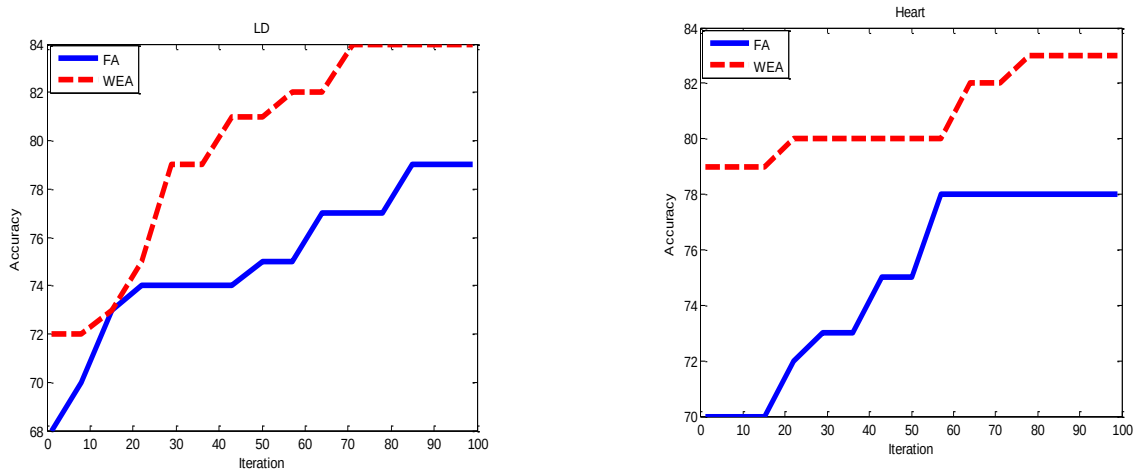


Figure 6. Convergence characteristics of WEA and FA.

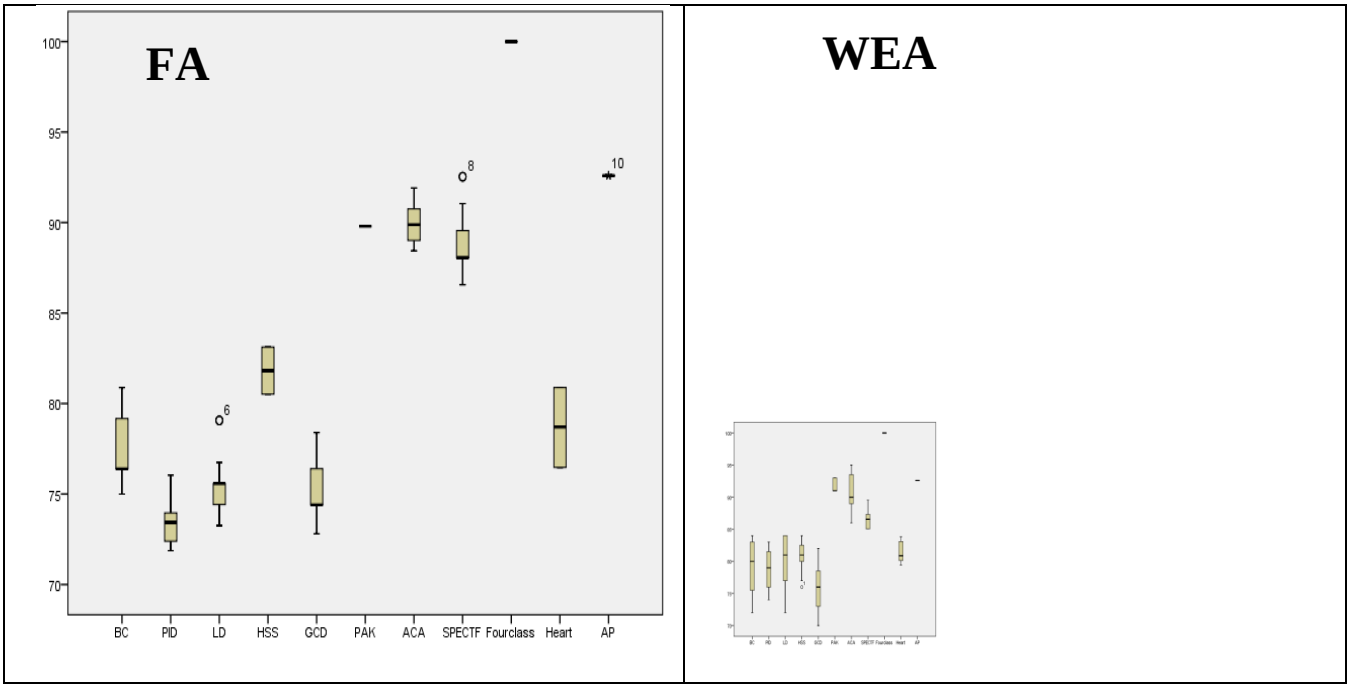


Figure 7. Box plots for FA and WEA.

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.

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

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.