

A Modified Particle Swarm Optimization with Neural Network via Euclidean Distance

<https://doi.org/10.3991/ijes.v6i1.8080>

Asia L. Jabar

University of Sulaimani, Kurdistan, Iraq

Tarik A. Rashid^(✉)

University of Kurdistan Hewler, Hewler, Kurdistan, Iraq

tarik.ahmed@ukh.edu.krd

Abstract—In this paper, a new modified model of Feed Forward Neural Network with Particle Swarm Optimization via using Euclidean Distance method (FNNPSOED) is used to better handle a classification problem of the employee's behavior. The Particle Swarm Optimization (PSO) as a Nature Inspired Algorithm is used to support the Feed Forward Neural Network (FNN) that has one hidden layer to gain the optimum weights and biases using different hidden layer neurons. The key reason of using Euclidean Distance (ED) with PSO is to take the distance between each two feature values and use this distance as a random number for the velocity value in the velocity equation in the PSO algorithm. The FNNPSOED is used to classify employees' behavior using 29 unique features. The FNNPSOED is evaluated against the Feed Forward Neural Network with Particle Swarm Optimization (FNNPSO). The FNNPSOED produced satisfactory results.

Keywords—Particle Swarm Optimization; Modified Particle Swarm Optimization; Feed Forward Neural Network; Euclidean Distance; Employee Behavior Classification.

1 Introduction

PSO was originally suggested by Eberhart and Kennedy in 1995 [1]. PSO is an optimization algorithm through which the bird swarm behaviors are represented. The PSO concept is inspired by the artificial life through imitating a population of flying birds. The population takes advantage of using collective collaboration among flying birds for arriving at the optimum value. PSO has been applied in numerous research areas such as optimization of functions, better training neural networks, optimizing parameters of other algorithms, the fuzzy system control, etc. A PSO particle in the algorithm is a flying bird. Normally, PSO can have several particles, these particles travel and transform their positions and their fitness is regularly move towards the best particle in the research area. The algorithm has an objective function (application-dependent) which controls the velocity and flying direction of particles [2]. It is

worth mentioning that PSO is part of evolutionary computational family. Accordingly, it is an iterative search algorithm. PSO has been widely used by researchers because of its simplicity, it is easily implemented, fast convergence, and contains few parameters. Nonetheless, PSO has the following defects [2, 3]:

1. It has a weak local search ability.
2. Its searching performance relies on parameters.
3. Its computational cost is expensive in some situations.

Particles in PSO tend to lose their capability for discovering new areas after searching in solution space (they fall into local optimization or what is called premature phenomenon).

However, it is very imperative that PSO can promise for converging towards the global optimum solution. Therefore, many researchers explored the standard PSO from various aspects such as topology building, introducing parameters, convergence, and hybridizing PSO with other swarm intelligence techniques. They pointed out the above mentioned limitations and then enhanced PSO to increase its performance to arrive at the global optimal solution [2, 3]. This paper aims at solving the PSO problems by detecting the distance between the values of each two features in the training dataset after summing these distances, the root of the result is found, and then the conclusive results are used instead of the random number in the velocity equation for the velocity value. This makes PSO more vigorous where there is no random number that may affect the result casually. Alternatively, a matrix of numbers can be determined from the used training dataset. Thus, in this research, a new modified FNNPSOED model is used and evaluated against a FNNPSO. FNNPSOED and FNNPSO are used for predicting what an employee will do in the future or to classify him/her according to their performance. Employers in the organizations must be able to distinguish talents and capacities of their employees, therefore, the classification task for the employee behavior is very imperative and valuable for decision purposes if the employee deserves the promotion or not.

The process of determining employee's performance in organizations in Kurdistan is executed manually through paper based works via managers at different sections or subsections. Accordingly, the effects of such a process remain typically objectionable as they are incorrectly judged.

2 Related Works

As it is mentioned that the goal of the PSO algorithm is to achieve the convergence and arrive at the global optimal solution, therefore, many research works focused on modifications on the PSO algorithm for the last ten years. In 1999, for instance, a technique was scrutinized to linearly reduce the inertia weight. They investigated the performance of PSO, for this purpose, 4 various standard functions with unequal initial range settings were preferred as part of testing functions. The research work established the capacity and incapacity of PSO. In all testing, PSO continuously converges so swiftly towards the ideal positions, yet, PSO gets slower when it is near enough

minimum. They concluded that using an adaptive inertia weight in their innovative approach enhanced the performance of PSO [4].

Del Valle et al. in 2008, presented the standard concepts of PSO and types of PSO in a summary. The research work produced a completed survey related to applications of power system. The research work recorded the influence of PSO on these applications and produced procedural specifics that are essential for implementing PSO. The research work also discussed the utmost effective fitness function, solution representation, etc. [5].

In 2009, an idea based on simulated annealing concept was introduced to modify PSO, the idea was done via two parameters (position and velocity) of flying birds. This modification reached greater steadiness and global convergence. The research concluded that particles with the minimum and maximum velocities would lead to superior influence on the convergence rate [6].

In 2012, Gunes and Ozkaya introduced a variant of the PSO algorithm. The research work was aimed to tackle multi-objective optimization tasks. The algorithm used minimum angular distance information to allocate the best local path for each flying bird. Also, the algorithm determined the field effect transistor model elements [7].

In 2013, a variant of PSO via simulated annealing technique was introduced by He and Guo. The model took the gradient descent approach as a particle swarm operator entrenched in PSO and simultaneously they used the damping technique with three input XOR problem for examining the enhancement of the PSO algorithm. Their results demonstrated that the ability of global optimization was increased and the premature convergence problem was avoided [8].

In 2014, Ding and Li developed a particle swarm optimization-Naive Bayes (PSO-NB). The model was a combination of both PSO and NB to choose features for enhancing the NB classifier. PSO in this model was used for obtaining the best subset of feature drop within the feature space, whereas, NB was used and implemented on the chosen subset of feature. The suggested model generated promising results [9].

In this research work FNNPSOED is used for classification; PSO was used to train the FNN classification technique to improve the classification rate and reduce the iteration number, while Euclidean distance (ED) is used to improve PSO. The dataset about employee behaviors and other details can be obtained at the following link [10]: “<https://github.com/asia-jabar/Employee-Inf-Dataset>”.

3 Theory Background

In this section, basic principles of the key algorithms are explained in detail.

3.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are regarded as computational representations that mimic biological neural networks [11]. The fundamental elements of ANNs are neurons. A standard structure of neural network consists of three layers (input, hidden

and output). Lower layers are connected to subsequent layers (higher layers) via connection weight or synapse. The net output of each neuron is produced via the summation of each input by its connection weights. Then, the net output is passed through a transfer function for producing its final output [12].

3.2 Particle Swarm Optimization

PSO is one of the key optimization methods that utilized for enhancing complicated or difficult arithmetical roles on representation of groups of fish or birds' social behaviors [1, 13]. In this algorithm, each solution of the optimization problem is like a bird or particle that searches through the pattern space [13] and each of which can fly from one place to another place in a search space of several dimensions. In the course of flying, a particle modifies its whereabouts based on its knowledge and the knowledge of other particles in its vicinity. As a consequence, a particle uses the best location came across by itself and its vicinity [1]. Each particle can be represented with four parameters (vectors). Let in D -dimension space, $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ represent the present position of the i^{th} particle, and $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, represent the best position initiated for the particle. Let also $n_i = (n_{i1}, n_{i2}, \dots, n_{iD})$ represent the best position initiated by its region and $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ represents both the direction of search as well as the speed with which the member/bird traverses the pattern space. The best position found by the particles during the search process is kept as p_{id} , besides, it also knows the best position n_{id} searched by the neighborhood particles, and changes its velocity [14]. The velocity and the position for each particle are calculated by using the following formula:

$$v_{id}^{k+1} = wv_{id}^k + c_1r_1(p_{id} - x_{id}^k) + c_2r_2(n_{id} - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (2)$$

where $i = 1, 2, \dots, N$, is the number of particles, $d=1, 2, \dots, D$, D is the dimension space; k , represents maximum number of iteration; $r1$ and $r2$ are useful for preserving the diversity of particles and they are randomly chosen values between 0 and 1; $c1$, $c2$ are the learning or acceleration coefficients; v_{id}^k is the velocity of the i^{th} particle with the d^{th} variable in k^{th} iteration in Eq. (1) [14]. Thus, the new position of the particle is represented by x_{id}^{k+1} , the previous position is represented by x_{id}^k , and the velocity by v_{id}^k .

3.3 Human Resources Management

Nowadays, Human Resource Management (HRM) in organizations is raised to a higher level and numerous procedures become important parts of HRM. Regardless of that, within HRM there are problems related to the decision making by the managers such as promoting employees, accepting new employees, and preventing employees from leaving the organization [10]. Thus, there might be reasonable that an experienced employee leave his/her organization. This will affect the performance of the organization in terms of quality and economy. Thus, artificial intelligence techniques

have been used to help decision makers to solve unstructured decisions in the applications of HRM. Data mining is regarded as the most common artificial intelligence technology that has been developed for discovering and examining huge sizes of data for defining relevant patterns [10, 15].

4 Proposed System Methodology

4.1 Euclidean Distance

Assuming that there is a space that has a collection of points and let function $d(x, y)$ be the distance measure on the space. This function can produce a number by taking two points as arguments in the space.

One common measuring distance technique is called Euclidean Distance. The most widely used formulae for Euclidean Distance can be expressed as:

$$d\left(\left[x_1, x_2, \dots, x_n\right], \left[y_1, y_2, \dots, y_n\right]\right) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3)$$

Where x_i is denoted as the cases in the first attribute and y_i is represented as cases in the second attribute.

4.2 Modified Particle Swarm Optimization

PSO in this paper is utilized for finding the optimum weight and biases for a neural network model. These weights and biases are then used for examining the model in the testing phase. The calculation of the anticipated optimization fitness function is conducted for each particle right after setting particles for PSO randomly. Afterward, all particles' position and velocity will be restructured based on their experience and other cohorts experience for arriving at the best position. In this research work a random number creator is suggested for PSO which would adjust the velocity of the particles. This is done through the Euclidean Distance formulae via which the distance between feature instances in each training case is determined and using it as a matrix of vectors for replacing the random number that exists in Eq. (1). After introducing the distance, Eq. (1) becomes as follows:

$$v_{id}^{k+1} = w^k v_{id}^k + c_1 \overset{\rightarrow}{ED} (p_{id} - x_{id}^k) + c_2 \overset{\rightarrow}{ED} (n_{id} - x_{id}^k) \quad (4)$$

Where $\overset{\rightarrow}{ED}$ is the normalized Euclidean distance value where its value gained by using Eq. (5). The normalized values are between 0 and 1 by using the following equation:

$$ED = (d - \min) / (\max - \min) \quad (5)$$

Where d is the calculated distance (through taking each two features as inputs using Eq. (3)), \min is the minimum value among the distance values, \max is the maxi-

imum value among the distance values. The generated \vec{ED} is used in Eq. (4) for updating the velocity of the particles with in the PSO algorithm.

Inertia Weight (w^k) in k^{th} iteration in Eq. (4) can also be updated using the following equation:

$$w^k = w_{max} - \frac{[Iteration_k * (w_{max} - w_{min})]}{Iteration_{max}} \quad (6)$$

w_{max} , w_{min} are the maximum and minimum values of Inertia Weights, and $Iteration_{max}$ is the maximum iteration number. Eq. (2) is used for updating the position of the particles in the dimension.

4.3 FNNPSOED

FNNPSO via ED is regarded as an updated version called FNNPSOED which is used for obtaining the distance between each two features in the training dataset. Thus, the obtained distance is used instead of a random number for updating velocity parameter in PSO.

Figure 1 represents the proposed model which consists of reading the training dataset which is used with calculating FNN and ED, then Eq. (5) is used for normalizing the distance from the previous step, v is the velocity of particle i at each iteration; pos is the position of each particle in the dimension of particles; $pBestScore$ represents a memory of the previous best position which is compared with the fitness for getting the $pBest$ value and updating the velocity of particles; $pBest$ is the best solution that is stored in the $pBestScore$ memory which is the cognitive component; $gBestScore$ is the memory of the best solution visited by any particle; $gBest$ represents the social components is the global best position which is the solution. $pBest$ and $gBest$ are the factors that helped in updating the velocity and position of the particles in the dimension and w is the Inertia Weight used to control the velocity.

5 Classification and Prediction

Basically, the methods of data mining might perform some significant tasks (for example, prediction, association rule, classification and clustering [16, 17]). Both artificial neural networks and particle swarm optimization are considered important methods that can solve various problems based on properties that they inherit. They used diverse ways for a minimization problem that might have one global minimum and more than a few local minima in the performance surface. Both FNNPSO and FNNPSOED are used for classifying employee behavior.

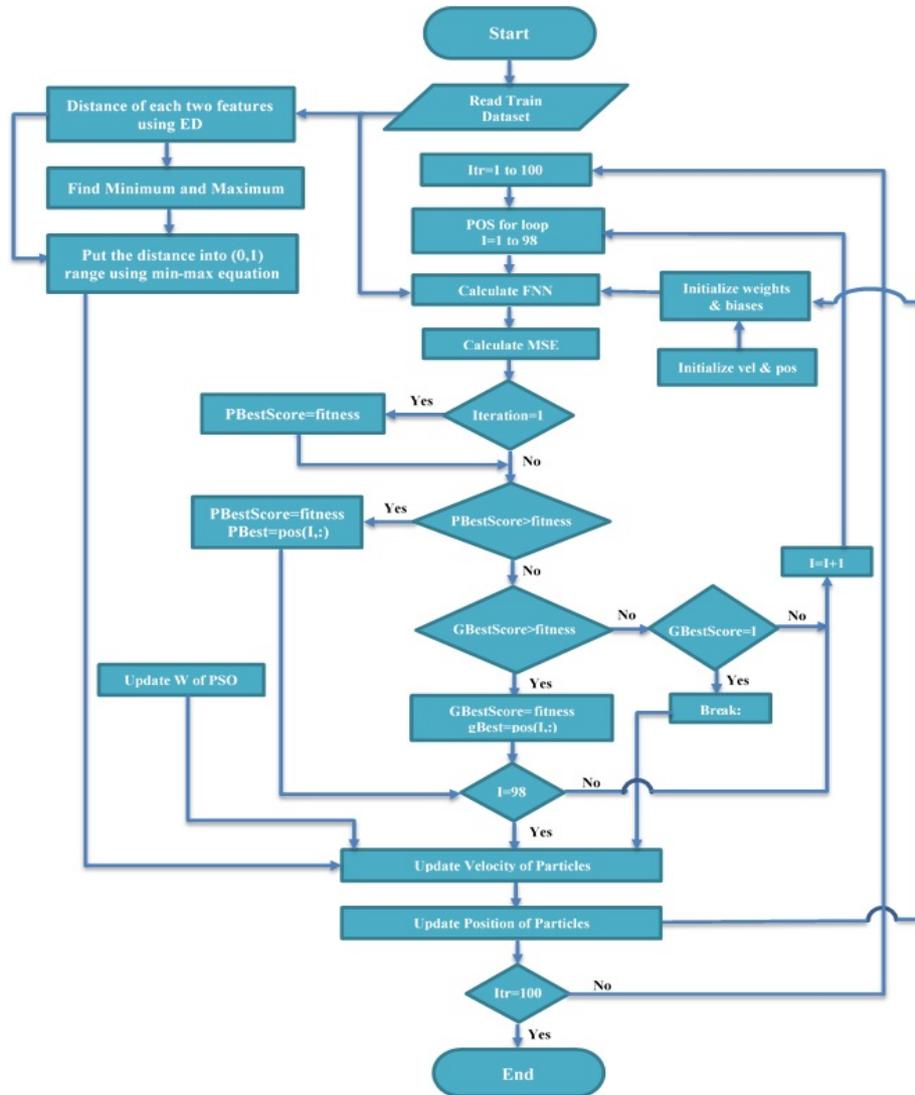


Fig. 1. The Proposed Model.

The dataset with 800 instances is used in the training session with 29 input features for determining the decision about an employee behavior whether the employee deserves a recommendation for promotion or not (the data sets have two class labels ; “yes” which means recommended for promotion, and “no” which means not recommended). Four types of datasets are used (350, 400, 500, and 600 instances) for the testing two experiments. Details of experiments are explained in the following subsections:

5.1 Experiment 1: Optimizing FNNPSO

Table 1 shows FNN parameters such as input features, hidden neurons, number of output class, maximum iteration number, and training number in the training session.

Number of particles (*NoP*), dimension of particles (*Dim*), Inertia Weight (*IW*), max Inertia Weight (*Wmax*), min Inertia Weight (*Wmin*), acceleration coefficient (*AC*), and momentum are the parameters of PSO which are represented in Table 2. These parameters are used with FNN for getting better weight and bias values with the goal of decreasing the error rate in the training session and getting higher classification accuracy rate.

The chosen *NoP* is 98, in fact, there isn't an apparent reason as to why we chose this number. It requires us to do numerous testing on PSO through altering population size for choosing the best size for this application. However, based on our experiments, it is hard to say what will be the population size, nonetheless, huge population size will produce precise result, but then, again tremendously large population size will increase memory space and the computational time.

Table 3 and 4 present the result of the confusion matrix in the training and the testing sessions. In each case, it can be clearly seen that the number of correctly and incorrectly classified instances is indicated.

More details about each performance measure as shown in Table 5 (Accuracy, Correctly Classified Instances (CCI) with its percentage, Incorrectly Classified Instances (ICI) and its percentage, Sensitivity, Fall-out, Specificity, MSE and others) are in [18]. The obtained results show that each case has good accuracy rate, but with error rate which affects the model in terms of classification.

Table 1. FNN Classifier Parameters

| Input Features | Hidden Neurons | Output Class | Training No. | Max Iteration |
|----------------|----------------|--------------|--------------|---------------|
| 29 | 35 | 2 | 800 | 100 |

Table 2. PSO Optimizer Parameters

| NoP | Dim | IW | Wmax | Wmin | AC | Momentum |
|-----|------|----|------|------|----|----------|
| 98 | 1998 | 2 | 0.7 | 0.5 | 2 | 0.8 |

Table 3. Confusion Matrix of Training Session

| Classified As | Actual | |
|---------------|--------|-----|
| | Yes | No |
| Yes | 436 | 13 |
| No | 5 | 346 |

Table 4. Confusion Matrix of Test Session

| Classified As | 350 Test Dataset | | 400 Test Dataset | | 500 Test Dataset | | 600 Test Dataset | |
|---------------|------------------|-----|------------------|-----|------------------|-----|------------------|----|
| | Yes | No | Yes | No | Yes | No | Yes | No |
| Yes | 44 | 16 | 74 | 25 | 130 | 55 | 557 | 13 |
| No | 25 | 265 | 2s2 | 279 | 25 | 290 | 14 | 16 |

Table 5. Evaluation Results of FNN-PSO

| Parameter | Training Session | Testing Session | | | |
|--------------|------------------|-----------------|---------------|---------------|---------------|
| | 800 instances | 350 instances | 400 instances | 500 instances | 600 instances |
| Accuracy | 0.97750 | 0.88571 | 0.88250 | 0.84000 | 0.95500 |
| CCI | 782 | 309 | 353 | 420 | 573 |
| CCI % | 97.75 % | 88.28 % | 88.25 % | 84.00 % | 95.50 % |
| ICI | 18 | 41 | 47 | 80 | 27 |
| ICI % | 2.25 % | 11.71 % | 11.75 % | 16.00 % | 04.50 % |
| Sensitivity | 0.97104 | 0.73333 | 0.74747 | 0.84680 | 0.97719 |
| Fall-out | 0.01424 | 0.08620 | 0.07309 | 0.17730 | 0.46667 |
| Specificity | 0.98575 | 0.91379 | 0.92691 | 0.82270 | 0.53333 |
| Miss-rate | 0.02895 | 0.26667 | 0.25253 | 0.15320 | 0.022807 |
| Precision | 0.98866 | 0.63768 | 0.77083 | 0.92401 | 0.97548 |
| Recall | 0.97104 | 0.73333 | 0.74747 | 0.84680 | 0.97719 |
| F-measure | 0.97977 | 0.68217 | 0.75897 | 0.88372 | 0.97634 |
| MSE | 0.02654 | 0.54922 | 0.53896 | 0.55353 | 0.42634 |
| Elapsed Time | 0.011161s | 0.0064138s | 0.00085002s | 0.00042449s | 0.00041294s |

Figures 4 and 5 described the obtained weights and biases which were used for training and testing the proposed model. Weight and bias rates in the figures mentioned are noticed to be very small or near zero. Clearly, this will not help the model to accept outliers and noises sensitively. There might be possibility of entering errors from users and this might lead to misclassification.

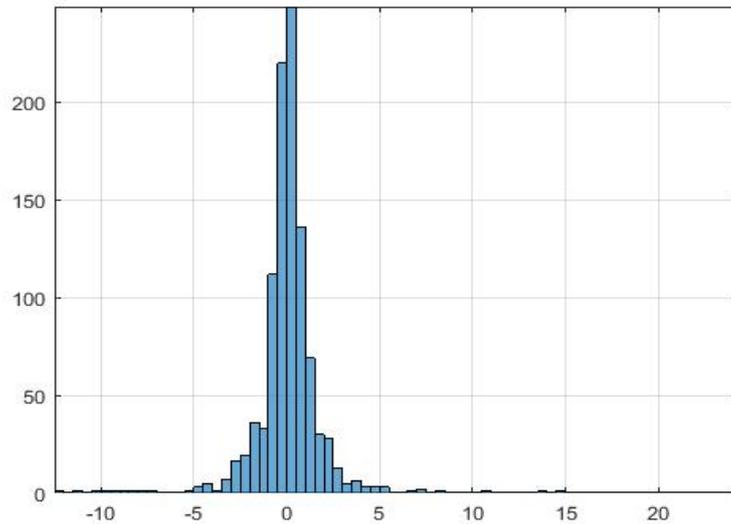


Fig. 2. Obtained Weights with FNNPSO

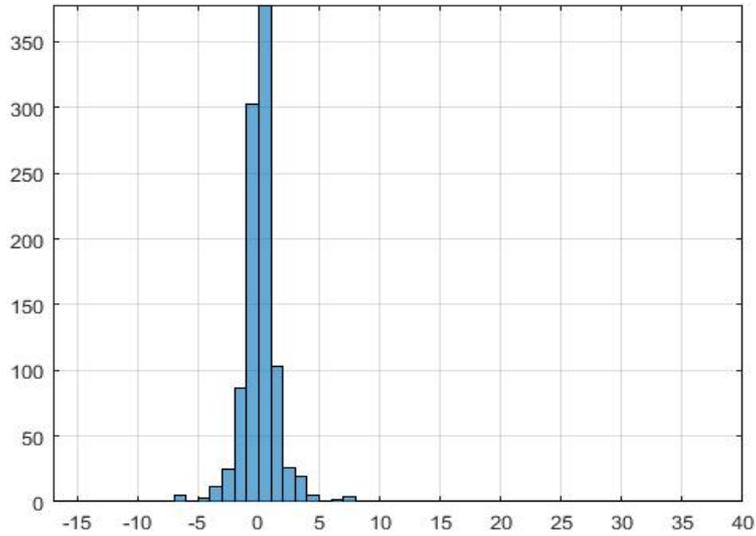


Fig. 3. Obtained Biases with FNNPSO

5.2 Experiment 2: Optimizing FNNPSOED

The FNNPSOED model is implemented by using the improved PSO with ED. This model attended to reduce error rate as much as possible by evaluating the improved PSO. Table 6, describes FNNs parameters and their values for implementing the proposed model.

Table (7), presents the improved PSOs parameters with their values that are used in the proposed system FNNPSOED.

Table 8, shows the confusion matrix results of the improved model in the training phase, where the performance of the model with the classification in each case is explained. The confusion matrix results in the testing phase are explained in Table 9.

Table 6. Proposed FNN Classifier Parameters

| Input Features | Hidden Neurons | Output Class | Training NO. |
|----------------|----------------|--------------|--------------|
| 29 | 37 | 2 | 800 |

Table 7. PSO Optimizer Parameters

| NoP | Dim | IW | Wmax | Wmin | AC | Min Position | Max Position | Max_Itr | Momentum |
|-----|------|----|------|------|----|--------------|--------------|---------|----------|
| 98 | 2112 | 2 | 0.7 | 0.5 | 2 | -0.6451 | 0.8259 | 100 | 0.8 |

Table 8. Confusion Matrix of Training

| Classified As | Yes | No |
|---------------|-----|-----|
| Yes | 349 | 9 |
| No | 4 | 438 |

Table 9. Confusion Matrix of Testing

| Classified As | 350 Test Dataset | | 400 Test Dataset | | 500 Test Dataset | | 600 Test Dataset | |
|---------------|------------------|-----|------------------|-----|------------------|-----|------------------|----|
| | Yes | No | Yes | No | Yes | No | Yes | No |
| Yes | 176 | 0 | 252 | 0 | 218 | 1 | 511 | 1 |
| No | 0 | 174 | 2 | 146 | 4 | 277 | 6 | 82 |

Evidently, as it is presented and explained in Table 8, the best results are with using the proposed model, where the higher rate of correctly classified instances and the lower rate of incorrectly classified instances are gained.

The proposed FNNPSOED evaluation results with the training and testing phases are described in Table 10. In this evaluation, it is noticed that the error rate, fall-out, and miss-rate reach the lowest values and the highest accuracy values are noticed.

Accordingly, few misclassification cases are found by comparing these two implemented models where in the first experiment which have higher error rates in each test cases. By comparing the elapsed time in these cases, it can be noticed that the FNNPSOED finishes with fewer time for one iteration.

Table 10. Evaluation Results of FNNPSOED (Training and Testing)

| Parameter | Training | | Testing | | |
|--------------|---------------|---------------|---------------|---------------|---------------|
| | 800 instances | 350 instances | 400 instances | 500 instances | 600 instances |
| Accuracy | 0.98125 | 1.00000 | 0.99500 | 0.99000 | 0.98833 |
| CCI | 787 | 350 | 398 | 495 | 593 |
| CCI % | 98.37 % | 100.0 % | 99.50 % | 99.00 % | 98.83 % |
| ICI | 13 | 0 | 2 | 5 | 7 |
| ICI % | 01.62 % | 0.000 % | 0.500 % | 01.00 % | 01.16 % |
| Sensitivity | 0.99101 | 1.00000 | 1.00000 | 0.99543 | 0.99805 |
| Fall-out | 0.03098 | 0.00000 | 0.01351 | 0.01423 | 0.06818 |
| Specificity | 0.96901 | 1.00000 | 0.98649 | 0.98577 | 0.93182 |
| Miss-rate | 0.00898 | 0.00000 | 0.00000 | 0.00456 | 0.00195 |
| Precision | 0.97566 | 1.00000 | 0.99213 | 0.98198 | 0.98839 |
| Recall | 0.99101 | 1.00000 | 1.00000 | 0.99543 | 0.99805 |
| F-measure | 0.98327 | 1.00000 | 0.99605 | 0.98866 | 0.99320 |
| MSE | 0.03719 | 0.29554 | 0.27841 | 0.25488 | 0.26962 |
| Elapsed Time | 0.010586s | 0.00041777s | 0.00040147s | 0.00041414s | 0.00038517s |

Different values of weights and biases are gained when FNN learned during the training of the proposed FNNPSOED. Figure 4 and 5 present the gained weights and biases values with this model. Unlike Figures 2 and 3, it can be seen from Figures 4 and 5 that the values of weights and biases are dispersed in parallel and are not gathered at one location, this means that the model would have a better chance to tackle noisy and outliers more softly and higher accuracy result is expected to be produced as well. Thus, this makes the model to be more robust and steady when dealing with the data sets in the testing cases.

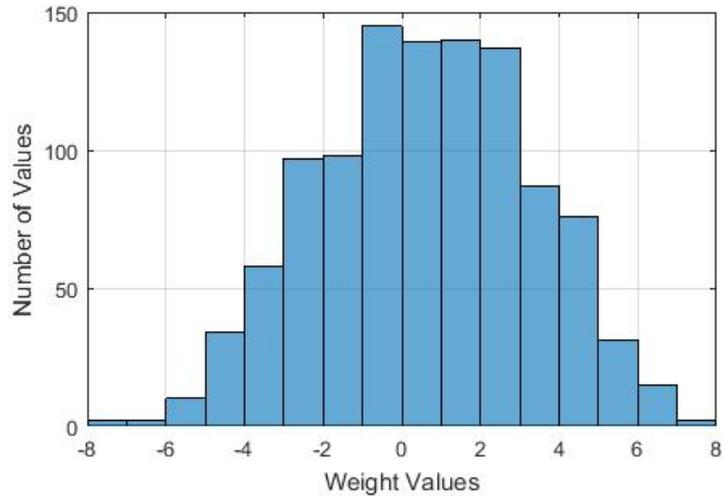


Fig. 4. Obtained Weights with FNNPSOED

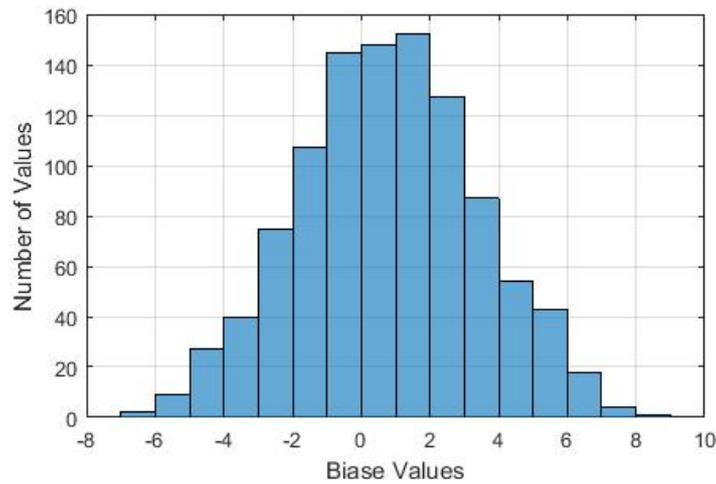


Fig. 5. Obtained Biases with FNNPSOED

In the two explained experimental models, decreasing the error rates is noticed with each training iteration that helped PSO to locate the best position for the particle that leads to the better results and attaining the goal for optimizing FNN Weights and Biases. The reduction of the error rate as mentioned in the two previous models can be noticed in Figure 6 where it is clear that the proposed FNNPSOED model has the lowest error rate in the second experiment which was started from 0.4 then after extended to its minimum value with each iteration, where both models started with the higher points. Another important point that must be mentioned is the number of iterations in reaching the lowest value of error rate where unlike the FNNPSO, the proposed FNNPSOED model reaches its lowest error rate value with lowest number of

iterations. Accordingly, it can be determined that the FNNPSOED needs less iterations to reach the goal. The decreasing in the error rate values between the two models can be illustrated in Figure 6.

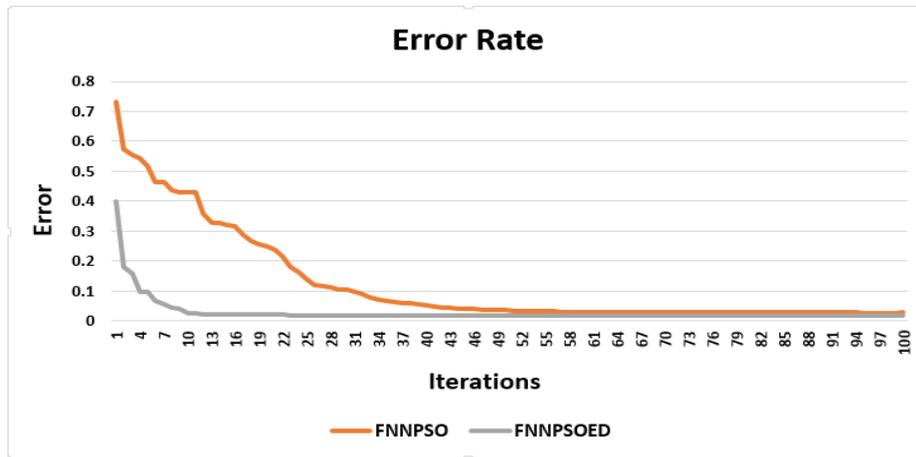


Fig. 6. Error Rate of Three Case

6 Conclusion

This research work tried to construct a model to replace the traditional ways in managing companies by using intelligent techniques in data mining for the decision makers to be able for deciding about their employees' behavior if they deserves to be promoted or not that. The working mechanism of the proposed model (FNNPSOED) was to generate a random number for PSO with FNN using ED method. Several points are concluded by building the proposed model. These points are concluded according to the various classification experimental results. Some of these points are summarized as follows: -

1. One important point for gaining the best weights and biases for FNN with one hidden layer is by using the nature inspired algorithms such as PSO for determining the best direction and the best position among particles according to the results.
2. The FNNPSOED received the highest classification accuracy rate and the lowest error rates via the ED technique (Euclidian probability distribution) for generating the random number within PSO.
3. The values of weights and biases in FNNPSOED model are wildly dispersed in parallel and are not gathered at one location, this means that the model would have a better chance to tackle noisy and outliers more softly and higher accuracy result is expected to be produced as well. Thus, this made the model to be more robust and steady when dealing with the data sets in the testing cases.

7 Future Recommendation

This research work can be extended into several ways as follows: -

1. Instead of using two classes, multiclass applications and other datasets can be used.
2. Other techniques of nature inspired algorithms such as Grey Wolf, Cuckoo Search and Artificial Bee Colony, etc. can be used instead of PSO.
3. Manhattan and Gaussian probability distribution techniques can be used as a substitution of Euclidian distance for generating random numbers in PSO.
4. Other types of ANNs such as Recurrent Neural Networks, Spiking Neural Networks and Deep Learning Neural Networks can be used instead of the used FNN.
5. Also, the number of the hidden layers can be changed and the memory of the selected network can be increased via adding more context layers as in Recurrent Neural networks.

8 References

- [1] Kennedy, J., and Eberhart, R. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. pp. 1942–1948.
- [2] Adnan, Akhtaruzzaman Md and Razzaque M. A. A comparative study of particle swarm optimization and Cuckoo search techniques through problem-specific distance function International Conference of. IEEE Information and Communication Technology (ICoICT), 2013.
- [3] Evers G.I. and Ghalia M. B., Regrouping Particle Swarm Optimization: A new global optimization algorithm with improved performance consistency across benchmarks, IEEE International Conference on Systems, Man and Cybernetics, 2009. <https://doi.org/10.1109/ICSMC.2009.5346625>
- [4] Shi, Y. and Eberhart, R.C. Empirical study of particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1945-1950. 1999. <https://doi.org/10.1109/CEC.1999.785511>
- [5] Del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., & Harley, R. G. Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Transactions on evolutionary computation, 12(2), pp. 171-195. 2008. <https://doi.org/10.1109/TEVC.2007.896686>
- [6] Mu A. Q., Cao D. X., and Wang X. H. A modified particle swarm optimization algorithm, Natural Science, 1(2), pp. 151-155, 2009. <https://doi.org/10.4236/ns.2009.12019>
- [7] Ozkaya, U., and Gunes, F. A modified particle swarm optimization algorithm and its application to the multiobjective FET modeling problem. Turk. J. Electr. Eng. Comput. Sci, 20(2), pp. 263-271, 2012.
- [8] Jie H, and Hui, G. A Modified Particle Swarm Optimization Algorithm, Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), 11(10), 2013.
- [9] Li, J., Ding, L., and Li, B., A Novel Naive Bayes Classification Algorithm Based on Particle Swarm Optimization, Open Automation and Control Systems Journal, 6, pp. 747-753, 2014. <https://doi.org/10.2174/1874444301406010747>
- [10] Jabar, A. L., and Rashid, T. A. Combining Fuzzy Rough Set with Salient Features for HRM Classification. In Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelli-

- gence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference, pp. 244-251, 2015. <https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.35>
- [11] Al-Radaideh, Q. A., and Al Nagi, E. Using Data Mining Techniques to Build a Classification Model for Predicting Employees Performance, *International Journal of Advanced Computer Science and Applications*, 3(2), pp. 144-151, 2012.
- [12] Singh, Y., and Chauhan A. S. Neural networks in data mining, *Journal of Theoretical and Applied Information Technology*, 5(6), pp. 36-42, 2009.
- [13] Somasundaram, P., & Muthuselvan, N. B. A Modified Particle Swarm Optimization Technique for Solving Transient Stability Constrained Optimal Power Flow, *Journal of Theoretical and Applied Information Technology*, 19(8), pp. 970-989, 2010.
- [14] Li, J., and Wang, X. A modified particle swarm optimization algorithm. In *Intelligent Control and Automation, WCICA 2004, Fifth World Congress 1*, pp. 354-356. IEEE, 2004.
- [15] Cheng, Z., and Chen, Y. Data mining applications in human resources management system. *Journal of Convergence Information Technology*, 7(8). 2012.
- [16] Leskovec, J., Rajaraman, A., and Ullman, J. D. *Mining of massive datasets*. Cambridge University Press, 2014. <https://doi.org/10.1017/CBO9781139924801>
- [17] Gupta, T., & Pandit, M. PSO-ANN for economic load dispatch with valve point loading effects. *Fuel*, 3000, 4000, 2012.
- [18] Sing, T., Sander, O., Beerenwinkel, N. and Lengauer, T. ROCR. Visualizing the performance of scoring classifiers. R package version 1.0-2, 2007. Accessed [19-08 2017]<ftp.auckland.ac.nz/software/CRAN/doc/packages/ROCR.pdf>

9 Authors

Asia L. Jabar is with Computer Science, University of Sulaimani, Kurdistan, Iraq.

Tarik A. Rashid is with Computer Science and Engineering Department, University of Kurdistan Hewler, Hewler, Kurdistan, Iraq.

Article submitted 07 December 2017. Resubmitted 09 December 2017 and 11 February 2018. Final acceptance 23 February 2018. Final version published as submitted by the authors.