WAS Science Nature
WASSN

ISSN: 2766 - 7715
WAS PRESS

Article

A New PSO-based ANN Hyper-parameters Selection Model

Hosam ALRahhal[1,2]*, Razan Jamous[1,3]

[1] Faculty of Engineering and Applied Science, University of Regina, Regina, Canada.
[2] Faculty of Engineering, NAHDA University, EGYPT, Hosamrahhal@gmail.com.
[3]Faculty of Computer Science, NAHDA University, EGYPT. razanadnanj@gmail.com.
*Corresponding author. E-mail address: Hosamrahhal@gmail.com; Tel: +1-639-999-4994.

Received: 01-04-2021; Accepted: 15-05-2021; Published: 20-06-2021

Abstract: Finding the best structure of ANN to minimize errors, the processing, and the search time is one of the main objectives in the AI field. In this paper, an enhanced PSO-based selection technique to determine the optimal configuration for the artificial neural network is presented. PSO with 2-D search space is used to select the optimal number of the hidden layer and the number of units per hidden layer. The proposed technique was evaluated using a chemical dataset. The result of testing the proposed technique displayed high prediction accuracy with a minimum error close to zero. In addition, the suggested technique reduces the mean absolute percentage error and mean absolute error significantly compared to ANN and PSO methods. Moreover, the relative error between the expected output and actual target is approximately between -0.02 and 0.02. The results of the comparison of the proposed technique with the ANN and PSO showed that the performance of the proposed approach is better in terms of the accuracy of the output prediction.

Keywords: Particle swarm optimization, Artificial Neural Network, Hyper-parameter, hidden layer.

1. Introduction

Due to the significant need for data prediction in different fields of life, Artificial Intelligence (AI) has recently attracted the attention of researchers. Among AI techniques, Artificial Neural Network (ANN) is considered one of the most important and effective technique. This technique has been used in various applications such as clustering [1], classification [2],[3], pattern recognition [4], and data prediction [5]. In particular, data prediction using ANN has become a familiar tool in many fields such as the oil industry, stock markets, computing, engineering, medicine, environmental, agriculture, and nanotechnology [6].

However, Hyper-parameters are defined as the parameters that have to be tuned before the beginning of the training process of the ANN. The hyper-parameters for the ANN technique include, for example, the number of hidden layers, the number of neurons, the activation function type, and the learning rate. The determination of the optimal hyper-parameters depends on selecting the suitable parameters that have to be tuned. After implementing the optimal value for each hyper-

parameter in an iterative way, the optimal hyper-parameters will be able to build the optimal architecture for ANN. This new model can improve the accuracy of future forecasted data. However, this procedure remains a challenging task.

Furthermore, the optimization of ANN performance is a critical issue. Although the ANN technique has the ability to reduce errors, this technique has many defects due to its dependency on hyper-parameters. In multilayer ANN (MLANN), if the complexity of ANN architecture (the hyper-parameters, for example, the number of hidden layers (HL) and the number of nodes per hidden layer (NPHL) is increased, then the requested number of training patterns, the processing time, and training time will increase. One way to improve the performance of this technique is the selection of the optimal hyper-parameters to predict future data with a high level of accuracy. Various methods have been suggested to optimize the structure of the multilayers neural networks, such as Particle Swarm Optimization (PSO) [7], back-propagation [8], genetic algorithms [9], ant colony optimization [10], bee swarm optimization (BSO) [11], Tabu search [12], and Fuzzy systems [13].

In this paper, a new PSO-based artificial NN configuration selection technique is proposed. The proposed technique concentrates on the problem of selecting the optimal configuration for ANN regarding the number of HL and the number of NPHL. In the selection process, PSO is used to tune the hyper-parameters of ANN.

The essential contributions of this paper can be summarized as follows:

1. Designing a generic technique that can be used to select the optimal hyper-parameters of an ANN.
2. Finding the optimal configuration for ANN in terms of the number of HL, and the number of NPHL using PSO for training ANN.
3. The proposed technique can be utilized to estimate the outputs of an ANN with high accuracy.
4. Comparing the proposed technique with traditional ANN.

The rest of the paper is organized as follows: a literature review is summarized in section 2. The description of the proposed technique is given in section 3. The performance evaluation is presented in section 4, followed by a conclusion and future work in section 5.

2. Related Work

The optimization of ANN architecture has been carried out by several researchers by tuning the hyper-parameters of ANN. This section summarizes the literature review in this area. For example, authors in [14], have proposed a Reservoir model for permeability prediction called HGAPSO in which the GA and PSO algorithms were integrated with the ANN to create a blended learning technique and optimizing the weights of the FFNN. In this method, the authors tested only five configurations with different numbers of hidden nodes to get the best configuration of artificial neural networks. The selection depends on the values of MSE, and the efficiency coefficient of the training and testing. The disadvantage of this study is the selected NN architecture was verified manually not automatically using a trial and error process. Furthermore, they selected a suitable structure from a few fixed numbers of structures, so the probability to select the optimal is decreased. The authors also used only one hidden layer, and they change only the number of hidden units per hidden layer. However, the proposed technique selected a suitable structure from all possible structures, increasing the probability of selecting the optimal structure. In addition, the proposed

technique used multiple HLs and multiple NPHLs. Furthermore, the proposed technique changed HL and NPHL to adjust selecting optimal structure.

In another study, authors in [15] proposed a new variant for PSO called cPSO-CNN to optimize the hyper-parameter settings for the selected architecture of the convolutional neural networks. A confidence function determined by a CND was used by the modified method for modeling the skills of researchers on the selection of configurations for convolutional neural networks for developing the exploration ability of cPSO. The modified technique also updates c1 and c2 of PSO as multidimensional vectors to best fit the various domains of hyper-parameters for convolutional neural networks. In addition, the authors used a linear estimation algorithm for speedy ordering the particles to increase the accuracy of this technique. It is worth noting that the authors of [15] used only the number of iterations as a hyper-parameter, without considering the architecture of the NN nor the number of HL or the nodes in every hidden layer. However, the proposed technique considered both NH and NPHL to select the best architecture of ANN.

An automatically-configuration finding method was presented in [16], to select the best network structure for DNNs using PSO and the steepest GD method. In this method, a set of scalar multidimensional vectors were used to represent network parameters as the particles of the PSO method in a search process. In the search process, a PSO method was used to find the best hyper-parameters of the network by moving the particles in a bounded search domain. To obtain a local optimal solution, the steepest GD algorithm was employed for training the deep neural network classifier with a small number of training iterations through the evaluation of PSO. After that, the steepest GD approach was completed with additional iterations and the best results of the particle swarm optimization technique for training a final model and individual deep neural network classifiers, respectively. The authors in [16], did not change the number of HL and the number NPHL at the same time. To evaluate their model, they ran two experiments with a fixed number for the hidden layers (two HL in the first experiment and three HL in the second experiment) and they changed the number of NPHL. However, the proposed technique used different values for both NH and NPHL that allow selecting the best hyper-parameters of ANN.

In [17], the authors proposed an automatic approach to optimize the hyper-parameters of CNN using tree growth and firefly algorithms. They applied their method for the image classification task. The simulation results proved that the proposed technique was robust, and achieved high classification accuracy. However, although this method is robust with high accuracy, it has a high computational cost while the proposed technique is efficient in terms of computational and time-consumption.

The authors in [18] presented a method that used PSO to select the parameters for deep learning models. To train deep learning models, they used a Wi-Fi dataset in order to estimate the number of occupiers and their locations. The simulation's results proved that the suggested PSO algorithm is efficient in training deep learning schemes. Also, the suggested PSO algorithm achieved higher accuracy in comparison with the grid search algorithm. In [18], the authors tested only selected constant numbers of HL and NPHL. However, their suggested method has a high computational cost. In addition, finding qualified parameter configurations consumed a long time. Compared to the suggested method, the proposed technique is efficient in terms of computational cost and consumption of time.

In [19], authors proposed automated methods to search for the best factors of CNN using PSO. The authors noted that the degree of convergence for convolutional neural networks training is affected by the selected dataset. As soon as the dataset is selected, using Spearman's ranking correlation it can determine the number of required iterations to train the network perfectly to guarantee fitness experiments. To decrease the expense of the training process, they presented an additional technique by introducing a stop criterion to measure the stabilization of the accuracy for the convolutional neural network. As soon as it becomes steady, the test of fitness will be stopped. The authors in [19], assuming that the network structure is already given with a constant number of HL and NPHL, attempted to optimize the related parameters using only the number of epochs as a hyper-parameter. Their method works to reduce the number of epochs to decrease the cost of determining the best hyper-parameter for convolutional neural networks without considering the change of the number of HL or the number NPHL while the proposed technique used different values for both NH and NPHL that allows selecting the best hyper-parameters of ANN.

In [20], the authors proposed a hybrid optimization technique called HPSOGA by combining PSO and GA. In their proposed technique, HPSOGA uses a fixed architecture network consisted of an input layer, one hidden layer, and an output layer. HPSOGA is utilized to determine the factors of RBF for NNs such as the number of nodes, their corresponding centers, and radius automatically. The authors employed the proposed algorithm to create the optimal components of RBF-NN that converge a function, including the number of hidden nodes, the centers, radius, and weights for them. HPSOGA is used in rainfall forecasting, the results display that the hybrid approach has more ability of global exploration and ability to avoid early convergence. The authors did not discuss the impact of the number of hidden layers on the performance of the algorithm. However, the proposed technique considered NH and used different values for NH that allows selecting the best hyper-parameters of ANN.

As can be seen from the discussion of the related literature, most of the proposed methods suffer from some problems such as using certain fixed numbers of hidden layers, a fixed NPHL, or the number of iterations. In addition, in most of these methods, the proposed approaches study the effect of only one of these hyper-parameters at the same time. Moreover, some of the suggested algorithms are time-consuming and have a high computational cost. In light of these limitations, we had the motivation to carry out this research.

3. The proposed Technique

In this section, the identification of the challenge and the suggested approach to overcome the identified challenge are described in detail.

The determination of the parameters of the neural network is more complex when large datasets with missing data are used. In addition, the percentage of uncertainty for requested data is very high when a large amount of data is lost and probably produces inaccurate results [21]. In this case, creating the appropriate architecture of the neural network that produces output with accepted error is very time and cost-consuming. To determine the architecture of the network, the configuration is generally carried out by hand, in a trial-and-error fashion. However, the manual tuning of the hyper-parameters of ANN through the trial-and-error process, and trying to find accurate configurations consume a long time. A different approach is the use of some global optimization techniques, for instance applying the PSO algorithm to choose the best architectures with minimum error. The hyper-

parameters such as the number HL and the number of NPHL have the highest priority in designing an ANN and also have a massive impact on the performance of ANN.

The purpose of the suggested technique is to solve the problem of designing the best structure for MLNN via selecting the best configuration for the network, i.e., selecting the best hyper-parameters to minimize the MSE and processing time.

Now, the standard PSO (SPSO) used in the proposed technique is described briefly. SPSO was presented by Kennedy and Eberhart [22].In principle, PSO mimics the simple behavior of organisms and the local cooperation with the environment and neighbor's organisms to develop behaviors that are used for solving complicated problems, such as optimization issues. PSO algorithm has many advantages compared with different Swarm Intelligence (SI) techniques. For instance, its search procedure is simple, effective, and easy to implement. In addition, it can effectively find the best global solutions with high accuracy. Moreover, PSO is a population-based search procedure in which each individual procedure represents a particle, i.e., a possible solution. These particles are grouped into a swarm. The particles moving within a multi-dimensional space, the particle's locations are adapted depending on its experience and that of its neighbors. The principle of the Particle Swarm Optimization technique can be explained [23] as follows. Let $X_i(t)=(x_{i1}^{(t)}, x_{i2}^{(t)}, \dots, x_{id}^{(t)})$ and $V_i(t) = (v_{i1}^{(t)}, v_{i2}^{(t)}, \dots, v_{id}^{(t)})$ denote the position and the velocity of a particle i in the search space at a time-step t, respectively. Also, let $P_i = (p_i1, p_i2, \dots, p_id)$ and $P_g = (p_g1, p_g2, \dots, p_gd)$ indicate the best solution established by the particle itself, and by the swarm, respectively. The new location of the particle is updated by adding a velocity to the existing position, as follows:

$$V_i^{(t+1)} = w.V_i^{(t)} + c_1 r_1\left(P_i - X_i^{(t)}\right) + c_2 r_2\left(P_g - X_i^{(t)}\right);\ i = 1,2,\dots.\mathrm{N}, \qquad (1)$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)};\ i = 1,2,\dots,N, \qquad (2)$$

Where the particle moving in a multidimensional space, c_1 and c_2 are positive constants, r_1 and r_2 are random numbers in the range [0, 1], and w is the inertia weight. $V_i(t)$ controls the optimization process, and denotes both the personal experience of the particle and swapped information from the other surrounded particles. The personal experience of a particle is usually indicated as the cognitive term, and it represents the approaching from the best local position. The swapped information is indicated as the social term in equation (1), which represents the approaching from the best global position for the swarm. In the proposed approach, two-dimensions search space was used for PSO. The first dimension is the number of HL and the second dimension is NPHL.

Any particle in the search space is considered as a candidate solution for the problem. This means that the location of the particle determines the values of NH and NPHL which represent a possible configuration for the network. In the search phase, the PSO technique is used to find the best settings by flying the particles within a bounded search space. Each particle has its own attributes which are location, speed, and fitness value calculated by a fitness function. The particle's speed defines the next movement (direction and traveled distance). The fitness value represents an index for the convergence of the particle from the solution. The position of each particle is updated to approach towards the individual that has an optimal location according to equations (1) and (2). In every repetition, every particle in the swarm modifies its speed and location based on two terms: the first is the individual optimal solution, which is the optimal solution that the particle can get personal. The second is the global optimal solution that is the optimal solution that the swarm can obtain cooperatively till now. Algorithm1 explains the pseudo-code of the suggested method. The

flowchart of the proposed approach is displayed in Figure 1. The suggested technique used a number of particles (pop) that are initialized randomly. Each particle must be selected based on its fitness, and the best solution is taken depending on the evaluation (performance) of each particle. The mean square error (MSE), which used as a fitness function in this paper, represents the error between the expected output and the actual output. In this case, the best particle is the one that generates the minimum MSE. The fitness function or objective function, which is a function that may assess how well or bad a position X is, that is, the quality of the solution. The fitness function maps the values in your particles to a real value that must reward those particles that are close to your optimization criterion. For each particle, the corresponding ANN is created and evaluated using a fitness function shown in equation (3).

$$f_i = \frac{1}{m} \sum_1^m (expectedoutput_i - actualoutput_i)^2 \quad (3)$$

where m is the number of inputs for ANN.

Algorithm 1: The proposed Algorithm

01: ***Begin***
02: Randomly initialize particles swarm
03: Run NN corresponding to each particle
04: ***while*** *(stopping standard is not met)*
05: ***for*** i *=1 to pop*
06: Compute the fitness value (f_i) *for each particle;*
07: ***if*** *the fitness value is less than* $Pbest_i$ *, i.e.,* (P_i) *in history;*
08: Set current fitness value as the new (P_i);
09: ***endif***
10: select the minimum (f_i) *of all particles in swarm as* $gbest_i$ *, i.e.,* (g_i);
11: ***for*** *d=1 to D*
12: Calculate $V_i^{(t+1)}$ *from Eq. (1);*
13: Calculate $X_i^{(t+1)}$ *from Eq.(2);*
14: ***end-for***
15: ***end-for***
16: return the best particle with gbest;
17: ***end-while***
18: Run the NN corresponding to the best selected hyper-parameters
19: ***end-algorithm.***

The fitness values of all swarm are determined using equation (3). The location and fitness value of each particle are stored as Xi and $Pbest_i$ respectively. Among all pbest, the pbest with minimum fitness value is selected as the global best particle (gbest) and thus the location and fitness value of gbest are stored. This process is repeated by updating the particle positions and velocities according to equations (1) and (2). The process is iterated until the best solution is found or the maxite is reached. The global best particle represents the best-selected hyper-parameters that used to build ANN.

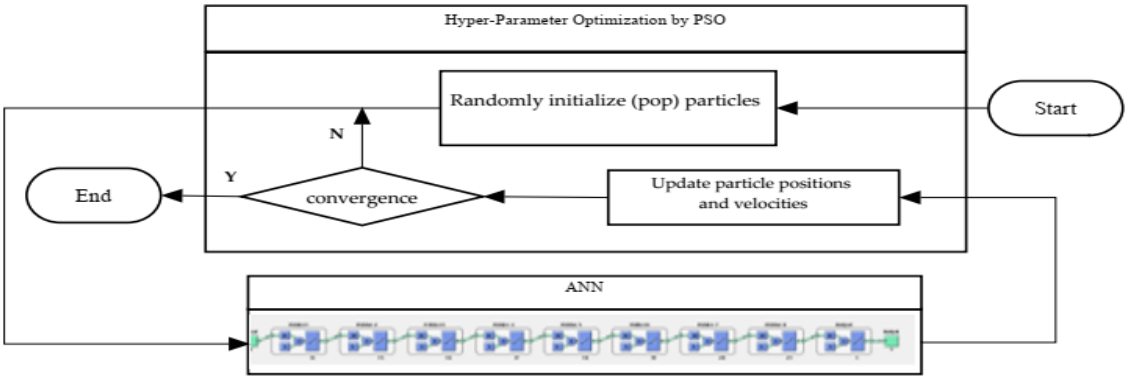


Figure 1. The flowchart of the proposed technique.

4. Experimental Results and Performance Evaluation

In this section, the evaluation of the performance of the new technique is presented. The experimental dataset and settings of the experiment are described in detail. In addition, the experimental results of the proposed approach are discussed.

4.1 Dataset description

In this subsection, the used datasets are described. To evaluate the suggested algorithm, the chemical sensor dataset was used. This dataset can be used to train a neural network to estimate one sensor signal from eight other sensor signals. The chemical dataset consists of an 8×498 matrix defining measurements taken from eight sensors during a chemical process used as inputs. In addition, the chemical dataset has a 1×498 matrix of a ninth sensor's measurements, to be estimated from the first eight sensors and used as targets. The output i.e. ninth sensor's measurement was reading daily. This dataset was used to evaluate the proposed method using mean square error and regression analysis. In this paper, while 80% of the available data were used as a training set, 10% of the available data were used as a validation set, and the last 10% of the available data were used as a test set.

The chemical sensor dataset can easily be loaded in the MATLAB workspace by typing the following commands:

```
[input, target] = chemical- dataset;
  inputs=input';
  targets=target';
```

4.2 Settings of experiment

To run the proposed method, the settings of PSO and multilayer neural network were set as follows: pop=20, maxite= 500, where pop is the swarm size while maxite is the maximum number of iterations. The values of c1, c2 were set to 2, and the values of Wmin, Wmax were set to 0.4 and 0.9, respectively [24],[25],[26]. The inertia weight (W) was linearly decreased from Wmax to Wmin [27]. We used a 2- dimensional search space (HL and NPHL) to represent the hyper-parameter of ANN. Initialization ranges for HL and NPHL were set to [1 7] according to [28], [29], and [14 21] according to [30], [31]., respectively. Levenberg-Marquardt (trainlm)[32] function was used as the train function

for the network. Tansig function was used as an activation function of the networks. In this paper, Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Relative Error (RE), and Mean Square Error (MSE) shown in Equation (4) to Equation (7) [33] are used to evaluate the accuracy of the proposed model.

$$MAPE=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{actualoutput_i-expectedoutput_i}{actualoutput_i}\right|\times 100\% \quad (4)$$

$$MAE=\frac{1}{n}\sum_{i=1}^{n}|actualoutput_i - expectedoutput_i| \quad (5)$$

$$RE_i=\frac{actualoutput_i-expectedoutput_i}{actualoutput_i} \quad (6)$$

$$MSE = \frac{1}{n}\sum_{1}^{n}(actualoutput_i - expectedoutput_i)^2 \quad (7)$$

4.3 Results and discussion

In this subsection, the results of applying the proposed technique are presented and discussed. In addition, the proposed technique is compared with traditional ANN without using PSO. To evaluate the performance of the suggested method, we modified the already existent MATLAB code of PSO and ANN packages according to our assumptions. To select the optimal configuration for the ANN, the proposed algorithm was tested in two different scenarios to determine the best settings for the proposed algorithm in terms of the population size and the number of iterations. In the first scenario, to show the effect of the population size on the proposed method, the population size was varied (pop= [10, 20, 30, 40, 50, 60,100]) while the maxite was kept constant for each population size. In the second scenario, the maximum number of iterations was varied (maxite= [25, 50, 100, 200, 300, 400, 500, 1000]) while the population size was kept constant at pop=20. The experiments were executed using a PC with Windows 8.1 operating system and Intel Core i5 processor running at 3.30 GHZ, 4 GB of RAM.

The result of the first scenario is presented in Figure 2. This Figure shows the variation in MSE against population size with the maxite equal to 1000, 500,300, 100, and 50. As can be seen from Figure 3, it is clear that the best population size with the minimum MSE is 20.

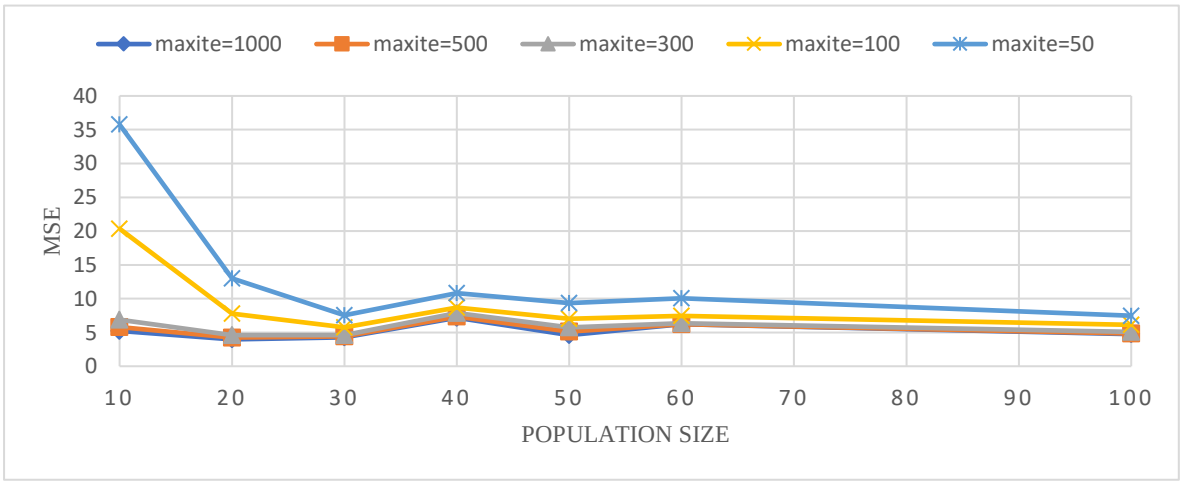


Figure 2. The variation in the MSE vs. the population size.

In the second scenario, to select the best maximum number of iterations the proposed technique was run for different values of iteration with a constant value for the population size (pop=20).

Figure 3 displays the relationship between the number of iterations and MSE.

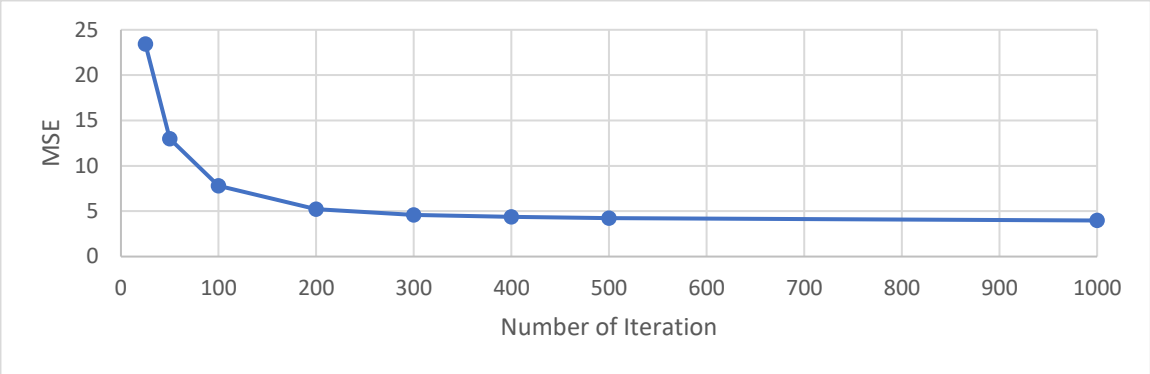


Figure 3. The relation between the number of iterations and the MSE.

It is clear that as the maximum number of iterations increases, the learning of the network increases, so as a result, the MSE decreases. It is also clear that the best number of iterations with the minimum MSE is 500.

Table 1, displays the elapsed time (sec) according to the number of iterations.

Table 1. The elapsed time (sec).

Population Size (pop) = 20	
Number of Iterations	**Elapsed Time(sec)**
25	17.86411
50	22.292923
100	31.592011
200	47.868706
300	64.288992
400	81.253257
500	110.071804
1000	175.033432

Now, after selecting the best value of the population size and the maximum number of iterations, the proposed algorithm was tested in three different scenarios to evaluate its performance.

Scenario #1: ANN was used.

Scenario #2: PSO was used.

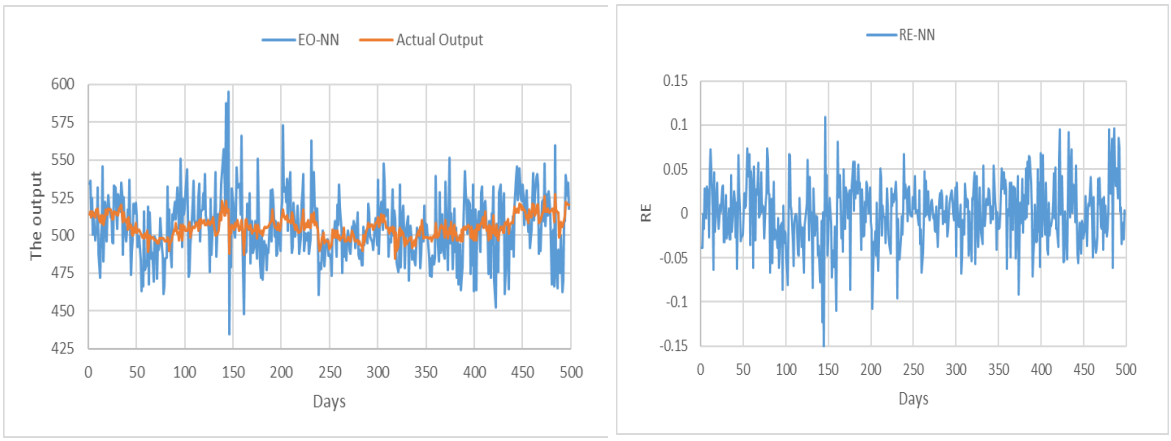
Scenario #3: The proposed technique was used.

Also, the proposed technique is compared with other models, i.e. ANN, and PSO.

4.3.1 Scenario #1

In this scenario, only the artificial neural network was used to predict the output. To evaluate this model, MAPE, MAE, and RE were calculated. The computed MAPE for this model is 2.85 %, while

MAE is 0.027. Figure 4 (a) shows the actual output (AO) and the Expected output for this model while Figure 4 (b) shows the relative error (RE) for the ANN model.

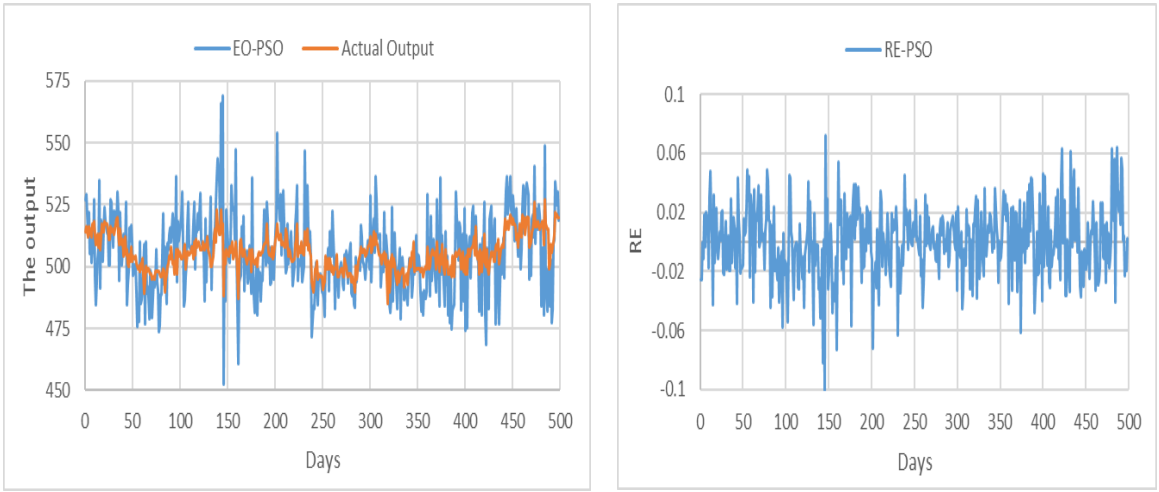


(a) AO vs. EO-ANN (b) The Relative Error for ANN model

Figure 4: ANN model performance.

4.3.2 Scenario #2

In this scenario, only Standard Particle Swarm Optimization (PSO) was used to predict the output. To evaluate this model, MAPE, MAE, and RE were calculated. The computed MAPE for this model is 1.90 %, while MAE is 0.018. Figure 5 (a) shows the actual output (AO) and the Expected output for this model while Figure 5 (b) shows the relative error (RE) for the PSO model.



(a) AO vs. EO-PSO (b) The Relative Error for PSO model

Figure 5: PSO model performance.

4.3.3 Scenario #3

In this scenario, a combination of ANN and the PSO (ANN-PSO) was used to predict the output, where SPSO was used to train the network to select the best configurations.

To design the best architecture of the ANN, the proposed technique was run with the selected settings to determine the best hyper-parameters of the network in terms of the number of HL and NPHL. The result of running the proposed technique displays that the best configuration was chosen by particle 10 with MPAE equal to 0.317 % and MAE equal to 0.002. As can be seen. the values of MPAE and MAE for the proposed model are very small and close to zero, thus the accuracy of the proposed technique is very high. The corresponding hyper-parameters, i.e., HL was equal to 2, the NPHL was equal to 15. Although obtaining the optimal set of hyper-parameters of the ANN is a time-consuming process [34], the elapsed time to find the optimal configuration in the proposed technique was 175.033432 seconds. Thus, the proposed algorithm can be considered efficient in terms of processing time. Furthermore, the regression (R) value measures the correlation between the predicted outputs and targets. When R=1 that means a close relationship whereas R=0 indicates a random relationship between the outputs and targets. Plot regression used to validate the performance of the proposed technique; besides that, it provides an important analysis of the results. The regression plots show the expected outputs of the network with respect to actual targets for training, validation, and test sets. Figure 6 shows the regression plot of the trained ANN with the selected hyper-parameters using the proposed algorithm. The regression coefficient R for training, validation, test, and all are 0.98236, 0.89948, 0.81845, and 0.94659, respectively. In addition, as can be seen, the fit line for data fall along a 45-degree line approximately which means that the network outputs are almost equal to the targets. Thus, the accuracy of the proposed method can be considered excellent.

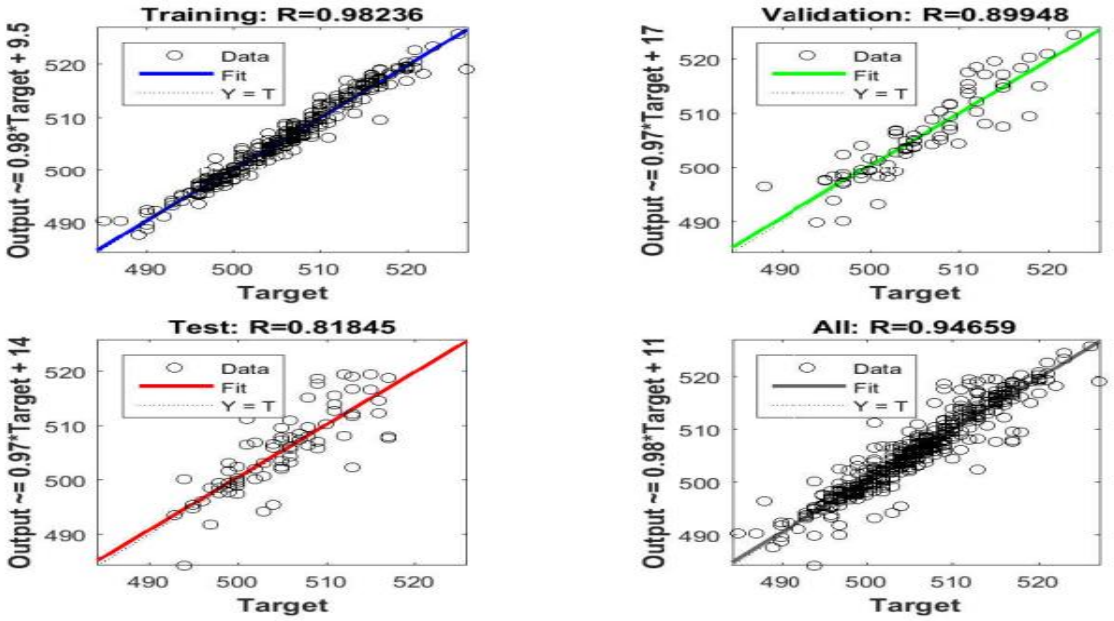
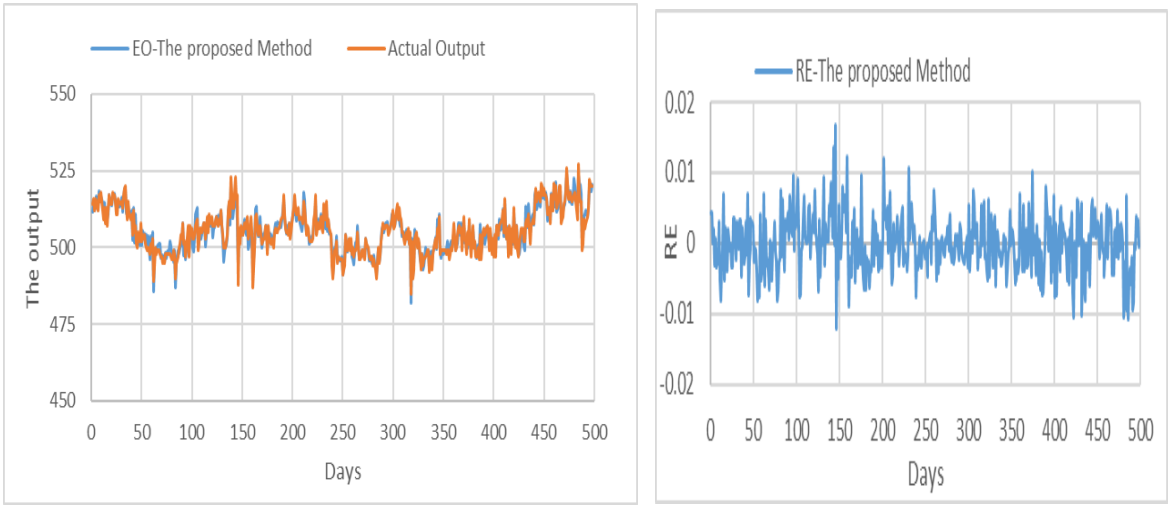


Figure 6. Regression plot of the trained ANN using proposed technique.

Figure 7 (a) shows the Actual output (AO) and the Expected output using the proposed method (EO-Proposed Method), while Figure 7 (b) shows the relative error (RE) of the proposed technique. As can be seen, the trained network using the proposed technique is giving the expected output with a relative error approximately between -0.02 and 0.02. Based on the above discussion, it is noted that the proposed model has a very high prediction accuracy.



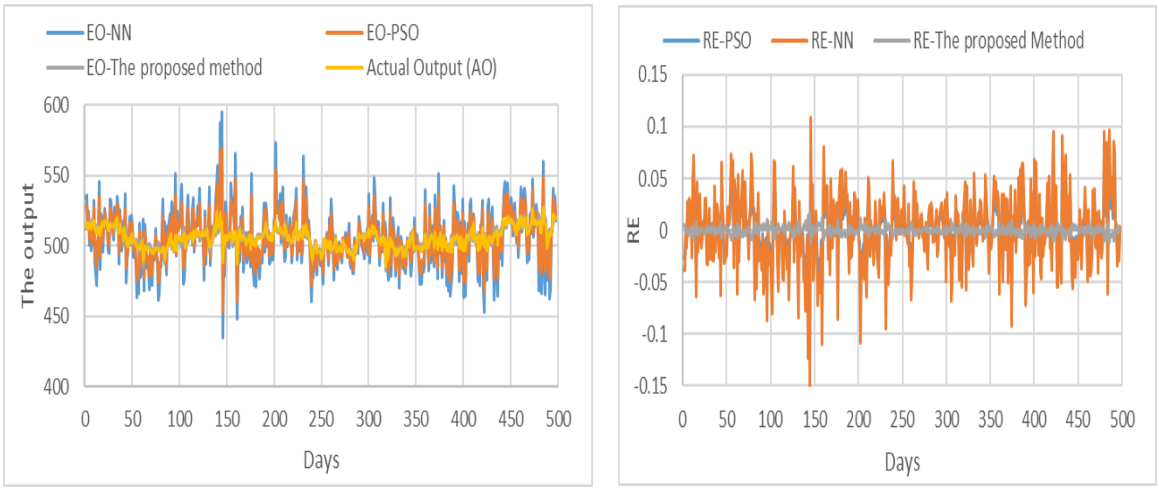
(a) AO vs. EO-The proposed Method (b) The Relative Error for The proposed Method

Figure 7: The proposed Method performance.

4.3.4 Comparison of the proposed model with ANN and PSO

Finally, to study the efficiency of the proposed technique, a comparison of this technique with the ANN and PSO was carried out. Figure 8 (a) shows the comparison between the expected output of the proposed method, the expected output of the PSO model, the expected output of the ANN model, and the actual output. While Figure 8 (b) shows the relative error (RE) for all models.

As can be seen from Figure 8 (a), it is clear that the convergence of the expected output of the proposed technique to the actual output with a very small error is better compared with the expected output of other models. Consequently, the proposed technique achieves higher accuracy than the ANN, and PSO models. As shown in Figure 8 (b), the RE of the proposed technique is significantly lower than the RE of other models. Therefore, the quality of the solutions provided by the proposed technique is higher than that provided by other models.



(a) The comparison between all models. (b) RE for all models.

Figure 8: performance of all models

To show the quality of the considering methods and their convergence to the best solution, MSE is used as a metric through the training phase. Figure 9 shows MSE for the compared models, it is noted the proposed method converged more rapidly than the remaining methods and achieved the best minimum value of MSE. This means that the prediction quality of the proposed method is the highest compared to other methods.

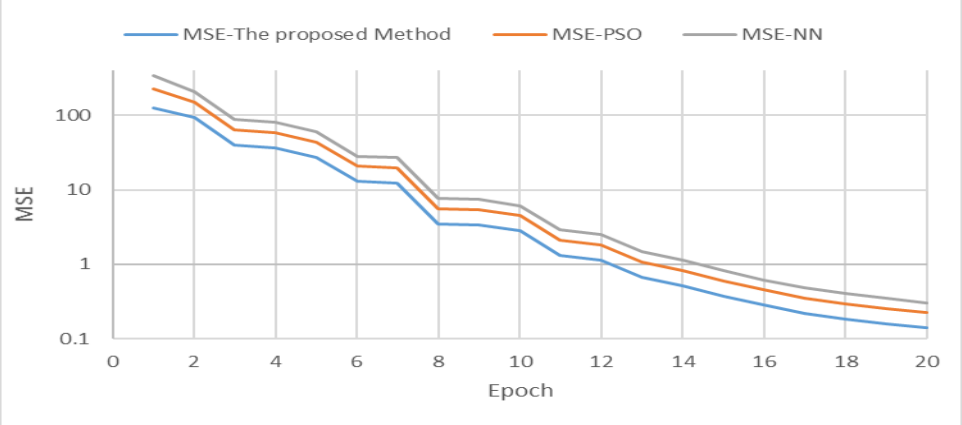


Figure 9: MSE of all models.

Table 2 shows MAPE and MAE for ANN, PSO, and the proposed methods.

TABLE2

MAPE, AND MAE, AND ET FOR ANN, PSO, AND THE PROPOSED METHODS

Metrics	ANN	PSO	the proposed method
MAPE%	2.85	1.90	0.317
MAE	0.027	0.018	0.002

As shown in Table 2, the proposed method performance is the best with the lowest values of MAPE and MAE compared to other models. By contrast, the ANN model displays the highest values of MAPE and MAE. Therefore, the proposed model's accuracy is the best against other models.

5. Conclusion and Future work

In this paper, a new technique is presented to select the best configuration for the Multilayer ANN using PSO. The proposed method is tested using a chemical dataset to evaluate the effectiveness of our method. The results show that the proposed technique is able to select the best hyper-parameters used to construct the desired network with very high accuracy of prediction with an error close to zero. Moreover, the proposed technique displayed better performance compared with ANN and PSO models in terms of the accuracy of the prediction.

For future work, we suggest the study of the effect of changing NPHL for each layer instead of being the same in all hidden layers in the proposed algorithm. We also suggest extending the proposed approach to discuss the ability to select more hyper-parameters such as the activation function, the number of iterations, the learning rate, and the size of a batch. In addition, future work may discuss the effect of the activation function if we apply the same activation function for all the layers or if we

apply a different activation function for each layer. Moreover, applying an improved PSO to train the NN and compare its results with the proposed algorithm is worthy of interest. Furthermore, a comparison of the proposed technique with other techniques such as GA, or ACO can be carried out. Finally, the proposed method can be applied against different datasets such as the oil reservoir dataset, COVID-19 dataset, historical data of electricity consumption, or stock markets dataset.

Author Contributions: Conceptualization, Razan Jamous, and Hosam ALRahhal; Methodology, Razan Jamous, and Hosam ALRahhal; Software, Razan Jamous and Hosam ALRahhal; Formal Analysis, Razan Jamous and Hosam ALRahhal; Data Curation, , Razan Jamous and Hosam ALRahhal; Writing-Original Draft Preparation, Razan Jamous and Hosam ALRahhal; Writing-Review & Editing, Razan Jamous and Hosam ALRahhal; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xu, D., Tian, Y. A Comprehensive Survey of Clustering Algorithms. Ann. Data. Sci. 2, 165–193 (2015). https://doi.org/10.1007/s40745-015-0040-1
2. Guoqiang P. Z. , "Neural Networks for Classification: A Survey ",IEEE transactions on systems, man, and cybernetics , Vol.30, Issue.4,451-462, 2000.
3. Liu, B.; Li, Y.; Li, G.; Liu, A. A Spectral Feature Based Convolutional Neural Network for Classification of Sea Surface Oil Spill. *ISPRS Int. J. Geo-Inf.* 2019, *8*, 160.
4. Devyani B. & Poonam S. Review on Reliable Pattern Recognition with Machine Learning Techniques, Fuzzy Information and Engineering, 10:3, 362-377,2018. DOI: 10.1080/16168658.2019.1611030
5. Xu, Z., Zhang, J., Wang, J. et al. Prediction research of financial time series based on deep learning. Soft Comput 24, 8295–8312 (2020). https://doi.org/10.1007/s00500-020-04788-w
6. Oludare A., Aman J., Abiodun O., Kemi D., Nachaat M., Humaira A. State-of-the-art in artificial neural networkapplications: A survey. Volume 4, Issue 11, November 2018,Heliyon 4 (2018) e00938. https://doi.org/10.1016/j.heliyon.2018.e00938
7. Zajmi L., Falah H. A, Adam A. J. "Concepts, Methods, and Performances of Particle Swarm Optimization, Backpropagation, and Neural Networks." Applied Computational Intelligence and Soft Computing, vol. 2018, 2018. https://doi.org/10.1155/2018/9547212
8. Hecht-Nielsen R. Theory of the backpropagation neural network. Neural Networks. 1988; 1(Supplement-1):445–448. https://doi.org/10.1016/0893-6080(88)90469-8
9. Islam BU, Baharudin Z, Raza MQ, Nallagownden P. Optimization of neural network architecture using genetic algorithm for load forecasting. In: 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS); 2014. p. 1–6
10. Ghanou Y, Bencheikh G. Architecture Optimization and Training for the Multilayer Perceptron using Ant System. International journal of computer science. 2016; 43(1):10 pages.
11. Ananthi J, Ranganathan V. Multilayer perceptron weight optimization using Bee swarm algorithm for mobility prediction. IIOAB Journal. 2016; 7(9):47–63.
12. Mustafa AS, Swamy YSK. Web service classification using multi-Layer perceptron optimized with Tabu search. In: Advance Computing Conference (IACC), 2015 IEEE International; 2015. p. 290–294.
13. Moayedi, H.; Tien Bui, D.; Gör, M.; Pradhan, B.; Jaafari, A. The Feasibility of Three Prediction Techniques of the Artificial Neural Network, Adaptive Neuro-Fuzzy Inference System, and Hybrid Particle Swarm Optimization for Assessing the Safety Factor of Cohesive Slopes. *ISPRS Int. J. Geo-Inf.* 2019, *8*, 391.
14. Mohammad A. Sohrab Z., Ali L, Ali E., Ioannis C. "Reservoir permeability prediction by neural networks combined with hybrid genetic algorithm and particle swarm optimization" ,Geophysical Prospecting, Volume 61, Issue 3, 2012.
15. Yulong W., Haoxin Z., Guangwei Z.," cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks" Swarm Evol. Comput. 49 (2019): 114-123.

16. Ye F. "Particle swarm optimization based automatic parameter selection for deep neural networks and its applications in large-scale and high-dimensional data". PLoS ONE 12(12): e0188746, 2017. https://doi.org/10.1371/journal. pone.0188746.
17. Bacanin, N.; Bezdan, T.; Tuba, E.; Strumberger, I.; Tuba, M. Optimizing Convolutional Neural Network Hyperparameters by Enhanced Swarm Intelligence Metaheuristics. Algorithms 2020, 13, 67
18. Qolomany B., Maabreh M., Al-Fuqaha A., Gupta A. and Benhaddou D., "Parameters optimization of deep learning models using Particle swarm optimization," 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, 2017, pp. 1285-1290.
19. Yamasaki T., Honma T., Aizawa K., Efficient optimization of convolutional neural networks using particle swarm optimization, in: IEEE Third International Conference on Multimedia Big Data, 2017, pp. 70–73.
20. Jiansheng W., JinL O., Mingzhe L., " Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm", in Neurocomputing Journal, Volume 148, 19 January 2015, Pages 136-142.
21. Frazier M, Longo C, Halpern BS (2016) Mapping Uncertainty Due to Missing Data in the Global Ocean Health Index. PLoS ONE 11(8): e0160377.
22. Eberhart R. C., Kennedy J., (1995). A new optimizer using particle swarm theory. In Proceedings of the sixth international symposium on micro machine and human science (pp. 39–43), Nagoya, Japan. Piscataway: IEEE.
23. Jamous R., Tharwat A., El. Seidy E., Bayoumi B., A new Particle Swarm with Center of Mass Optimization, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 04, Issue 05 (May 2015).
24. Shi Y, Eberhart R.. Empirical study of particle swarm optimization. International Conference on Evolutionary. Computation. Washington, USA, 1999: 1945-1950.
25. Xin J., Chen G. and Hai Y., "A Particle Swarm Optimizer with Multi-stage Linearly-Decreasing Inertia Weight," 2009 International Joint Conference on Computational Sciences and Optimization, Sanya, Hainan, 2009, pp. 505-508, doi: 10.1109/CSO.2009.420.
26. Sengupta, S.; Basak, S.; Peters, R.A., II. Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives. Mach. Learn. Knowl. Extr. 2019, 1, 157-191.
27. Sengupta S., Basak S., Peters R. A., "Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives" Mach. Learn. Knowl. Extr. 1, 157-191, 2019.
28. Qiu M., Song Y., Predicting the Direction of Stock Market Index Movement Using an Optimized Artificial Neural Network Model. PLoS ONE 11(5): e0155133, 2016.
29. Panchal G., Ganatra A., Kosta Y., Panchal D., "Behaviour Analysis of Multilayer Perceptrons with Multiple Hidden Neurons and Hidden Layers", International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011.
30. Thomas A. J., Petridis M., Walters S. D., Gheytassi S. M., and Morgan R. E., "On predicting the optimal number of hidden nodes," in Proc. 2015 International Conference on Computational Science and Computational Intelligence), 2015, pp. 565–570.
31. Thomas A. J., Walters S. D., Gheytassi S. M., Morgan R. E., and Petridis M., "On the Optimal Node Ratio between Hidden Layers: A Probabilistic Study," International Journal of Machine Learning and Computing vol. 6, no. 5, pp. 241-247, 2016.
32. Moré J. J., "The Levenberg-Marquardt algorithm: Implementation and theory," in Numerical Analysis, G. A. Watson, Ed. Springer Berlin Heidelberg, 1978, pp. 105–116.
33. Botchkarev A., "A new typology design of performance metrics to measure errors in machine learning regression algorithms," Interdisciplinary J. Inf., Knowledge & Management, vol. 14, pp. 45–76, 2019.
34. Stang M., Meier C., Rau V., Sax E. (2020) An Evolutionary Approach to Hyper-Parameter Optimization of Neural Networks. In: Ahram T., Taiar R., Colson S., Choplin A. (eds) Human Interaction and Emerging Technologies. IHIET 2019. Advances in Intelligent Systems and Computing, vol 1018. Springer, Cham.