International Journal of Machine Learning and Cybernetics (2021) 12:1517–1544
https://doi.org/10.1007/s13042-020-01252-x

ORIGINAL ARTICLE

Vortex search optimization algorithm for training of feed-forward neural network

Tahir Sağ[1] · Zainab Abdullah Jalil Jalil[1]

Received: 7 August 2019 / Accepted: 1 December 2020 / Published online: 4 February 2021

Abstract
Training of feed-forward neural-networks (FNN) is a challenging nonlinear task in supervised learning systems. Further, derivative learning-based methods are frequently inadequate for the training phase and cause a high computational complexity due to the numerous weight values that need to be tuned. In this study, training of neural-networks is considered as an optimization process and the best values of weights and biases in the structure of FNN are determined by Vortex Search (VS) algorithm. The VS algorithm is a novel metaheuristic optimization method recently developed, inspired by the vortex shape of stirred liquids. VS fulfills the training task to set the optimal weights and biases stated in a matrix. In this context, the proposed VS-based learning method for FNNs (VS-FNN) is conducted to analyze the effectiveness of the VS algorithm in FNN training for the first time in the literature. The proposed method is applied to six datasets whose names are 3-bit XOR, Iris Classification, Wine-Recognition, Wisconsin-Breast-Cancer, Pima-Indians-Diabetes, and Thyroid-Disease. The performance of the proposed algorithm is analyzed by comparing with other training methods based on Artificial Bee Colony Optimization (ABC), Particle Swarm Optimization (PSO), Simulated Annealing (SA), Genetic Algorithm (GA) and Stochastic Gradient Descent (SGD) algorithms. The experimental results show that VS-FNN is generally leading and competitive. It is also said that VS-FNN can be used as a capable tool for neural networks.

Keywords FNN · Classification · Optimization · Training neural-networks · Vortex search

1 Introduction

Artificial Neural Networks (ANNs) are one of the most useful computational modeling tools in machine learning algorithms. Therefore, ANN has attracted attention in many disciplines including engineering, medicine, agriculture, technology, business, arts, etc. Up to now, it has been applied as a reliable method to various challenging manners such as classification, image processing, speech recognition, natural language processing, etc. ANN supplies a parallel computing system including many simple processors inspired by biological neural networks and uses some organizational principles used in humans. In ANN, the connection weights storing information are combined in a parallel and sequential manner which forms the network architecture. In many real-world implementations, feed-forward neural networks (FNNs) are the most popular neural networks.

The concept of learning refers to the process of finding optimal weight values in this architecture. The success of ANNs in problem-solving depends largely on the training of the network and the performance of the learning algorithm used in the training phase [1, 8, 13, 23, 24]. Several learning algorithms have been used in literature. The most popular training methods are based on mathematical error reduction techniques such as back-propagation (BP) [6], Gradient Descent (GD) [29], Conjugate Gradient [36], Newton's Method [5] and Levenberg–Marquardt (LM) [12]. Due to some factors such as the problems having nonlinear properties and/or large-scale dimensions, derivative learning-based algorithms may not always be sufficient for training neural networks.

Many metaheuristic optimization algorithms have been widely used in solving NP-hard problems such as training of FNNs since these algorithms establish a balance-aware between exploration and exploitation so that providing the optimal solution in the search space [17]. Some of the

✉ Tahir Sağ
tahirsag@selcuk.edu.tr

[1] Department of Computer Engineering, Selcuk University, Konya 42075, Turkey

most-cited algorithms developed to date are Artificial Bee Colony (ABC) [16], Differential Evolution Algorithms (DEA) [32], Genetic Algorithms (GA) [14], Simulated Annealing (SA) [19], Particle Swarm Optimization (PSO) [18], Grey Wolf Optimizer [21], Firefly Algorithm (FA) [40], Butterfly Optimization Algorithm [3], Biogeography based Optimization Algorithm [30] and Gravitational Search Algorithm [28]. Apart from these, there are more than 250 metaheuristic algorithms presented in the literature for various purposes based on the NFL theorem [37].

This paper focuses on Vortex Search (VS) optimization algorithm [9] and it is adapted to the training task of FNNs as a learning method. The VS algorithm has been recently proposed for solving numerical function optimization and it uses multivariate Gaussian distribution to generate candidate solutions. The search space is reduced by using the inverse incomplete gamma function during iterations. Thus, exploration is effectively accelerated while performing exploitation.

To the best of our knowledge, this is the first study that the efficiency of VS is investigated for the training of FNNs and a VS-based learning method for FNNs (VS-FNN) is proposed in this paper. In order to adapt the VS algorithm to the training phase, the training process is accepted as an optimization problem and all weights and bias values in the architecture of FNN are systematically stored in a matrix to generate and optimize as a candidate solution in the search space. In other words, each weight value in this matrix represents a component of the center point for the vortex in the algorithm.

1.1 The motivation and contribution

As reported in the paper [9], VS was tested over 50 benchmark mathematical functions and the results were compared to both the single-solution based (SA and Pattern Search) and population-based (PSO and ABC) algorithms. Herewith, it was reported that VS outperforms or at least competed with other algorithms. This report proves that VS ensures the balance strongly between exploration and exploitation. Our motivation at the beginning of this work is based on the fact that it has not been previously investigated in the training of FNN, although it has been stated to be very successful against the most advanced algorithms.

The efficiency of the VS algorithm is examined in the training of FNN by comparing it with ABC, PSO, SA, GA, and SGD algorithms. However, it is not within the scope of this study to find the optimal network architecture to provide the lowest possible training error. The aim is to demonstrate that the VS algorithm is as competitive as other metaheuristics in training neural-networks and to analyze its performance. To crosscheck the accuracies of the trained FNNs, the algorithms are run over six datasets with multiple classes whose names are called 3-bit XOR, Iris Classification, Wine-Recognition, Wisconsin-Breast-Cancer, Thyroid-Disease, and Pima-Indians-Diabetes. In summary, the contributions of VS-based learning algorithm can be bulleted as follows:

- To make the first investigation effort for the VS algorithm in the training of FNNs.
- To demonstrate the effectiveness of its running nature not needing a user-tuning parameter on the adjustment of weights in an FNN.
- To prove that it can achieve an accuracy performance as high as at least other successful algorithms by consuming less computation time.

The rest of the paper is organized as follows: some previous related works are introduced in Sect. 2. Then, the definition of FNN is explained briefly in Sect. 3. Vortex Search optimization algorithm is described in Sect. 4. After that, the VS-based learning method is revealed in Sect. 5. The experimental results are demonstrated and evaluated in Sect. 6. Finally, the conclusions are briefly considered in Sect. 7.

2 Related works

The focus here is to adjust the optimal weights and biases in an FNN structure. In this way, training neural networks can be accepted as an important nonlinear optimization problem. In literature, besides gradient methods, a large number of metaheuristics have been presented to train neural networks. Some of the mostly-used algorithms depend on evolutionary techniques and swarm intelligence algorithms.

One of the first studies in this field was presented by [25]. They used a GA-trained FFN to classify data of a sonar image dataset. They reported that GA could achieve better results than BP in particularly domain-specific problems. In 1998 [35], revealed a training method by combining gradient descent with SA. They claimed to maintain a quick convergence without tackling local optima during the training process.

In the literature, there are many papers presented for the training of FNN with PSO. For example, in 2007, (Gudise and Venayagamoorthy 2003) made a comparative study between BP and PSO for training neural networks. Considering computational requirements, they concluded that PSO had a faster convergence than BP. Then [41], presented a hybrid learning algorithm combining PSO with BP for the same purpose. They took advantage of the global search ability of PSO at the beginning of the iterations and later used BP for local search. They used three benchmark problems (3-bit-parity, function approximation, and Iris classification) and reported that their proposed method had reached better convergence than PSO and BP. In another study [31], adapted ant colony optimization to continuous optimization and they applied it to the training of feed-forward neural networks. The obtained results were evaluated as successful as GD by the authors [15]. used BP and PSO for training neural networks. In order to solve medical diagnostic problems, the performance between global and local techniques was evaluated on the training. They analyzed the performance of the algorithms on three medical diagnostic problems stated in the Proben1 dataset. BP was found to be more efficient than PSO [39] trained neural-networks with a combined method involving opposition based PSO and BP with momentum. They examined the hybrid method on well-known eight datasets. The produced results for training time and classification accuracy showed the superiority of the method.

ABC is another metaheuristic proposed for FNN training in literature [26] designed ABC algorithm to solve XOR, 3-Bit Parity, Encoder-Decoder test problems, and classification problems. The performance of the ABC algorithm for training neural networks was compared with BP, LM, GA, and PSO algorithms. They were reported that the ABC-based ANN training method reached more successful results by obtaining the highest classification success. (Brajevic and Tuba 2013) are used FA to train neural-networks by using two different transfer functions: sigmoid and sine. They also investigated the performance of FA with the results of ABC and GA obtained from benchmarks XOR, 3-Bit Parity, and 4-bit parity. According to experimental results, they concluded that superiority order among three algorithms was ABC, FA, and GA, respectively. A new study for FNN training with ABC was recently proposed by [38]. They modified ABC to accelerate the convergence and added a new selection method to improve the performance. Then they compared the training results with ABC variants.

Some of the various meta-heuristic optimization algorithms recently proposed as the learning method for FNNs are mentioned below. For example, Mirjalili et al. [22] focused on the Gravitational Search Algorithm (GSA) for training FNN and they proposed a hybrid learning method (PSOGSA) combining GSA and PSO. They compared the hybrid method with the original GSA and PSO by applying on three benchmark problems. Consequently, PSOGSA achieved better results in terms of converging speed and accuracy.

In another recent study, bird mating optimizer (BMO) inspired from the behaviors of birds in mating time, was used by [4] for the training of neural networks. They said that promising results had been achieved by the BMO-based training method on three benchmark datasets: Iris, Wisconsin-Breast-Cancer, Pima-Indian-Diabetes, and fuel cell system.

Piotrowski [27] had researched the performance of previous DEA based training methods and the researcher thought that stagnation is the main reason why it fails compared to other methods. In order to overcome this problem, it is reported that he achieved satisfactory results by combining global and local neighborhood-based mutation operators with the trigonometric mutation operator.

Tang et al. [34] adapted Dynamic Group Optimization (DGO) algorithm for training FNN. They used the DGO algorithm to find the optimal weights and bias values as well as to determine the structure of the FFN. Considering the reported experiments, they claimed that DGO is a suitable method for FNN training.

Swain et al. [33] proposed a hybrid metaheuristic combining the Gravitational Search algorithm and PSO for training FNN to diagnose the faults in wireless sensor networks.

3 Feed-forward neural-network

Feed-Forward Neural Network (FNN) is the most popular type of neural-network in which the data flow occurs only in the forward direction since FFNs can solve classification and regression problems effectively. The basic processing unit in FFN is defined as a neuron, which is inspired by biological nerve cells. Figure 1 shows the structure of an artificial neuron. Every single neuron generates one output signal corresponding to an input vector with n-dimensional. To accomplish a particular task, neural networks are trained by updating the values of the connections between neurons. These connections are usually named as weights and the connections do not form a cycle.

In FNN, the neurons are grouped in three types of layers: (i) input layer containing an equal number of neurons with the same number of problem-inputs, (ii) hidden layer containing at least one neuron that has to be in the solution of non-linear NP-hard problems and (iii) output layer containing an equal number of neurons with a number of problem-outputs. The numbers of neurons in input and output layers depend on the problem, while the number of

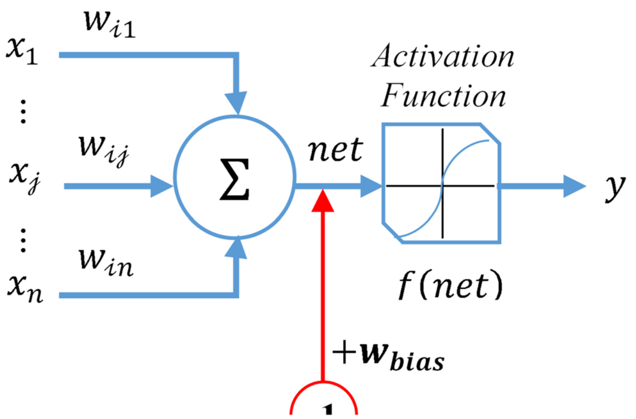

Fig. 1 The structure of a neuron

neurons in the hidden layer and the number of hidden layers are defined by decision-makers who set the architecture to be applied [2]. The schematic of a general three-layer FNN is shown in Fig. 2.

Firstly, the weighted sum of inputs is calculated by Eq. (1) and then output *(y)* is calculated by sigmoid function as follows in Eq. (2).

$$net = \sum_{\rangle=1}^{n} w_{\rangle|}x_{|} + w_{bias}, \tag{1}$$

$$y = f(net) = \frac{1}{1 + e^{-\mathrm{net}}}, \tag{2}$$

where $x_|$ indicates the i.th input and $i \in \{1,2,\ldots,n\}$. $w_{\rangle|}$ is the weight between i.th input and j.th neuron. The red input with constant one value in Fig. 1 is a bias which is the weight value of a neuron which is independent of the inputs and it is applied to the hidden and output layers in order to balance the weight values in the training phase. The output of each neuron is calculated by a transfer function such as linear, radial, sigmoid, and hyperbolic tangent.

The aim of the learning procedure is to minimize the error caused by the difference between the calculated and expected output values. ANN is trained by inputs in training set until the output of the network satisfies the expected output. Then output values for unknown input values are estimated by using a trained network. To obtain acceptable results from the complex and nonlinear multi-variable problems, ANN should be trained with sufficient data using an appropriate learning algorithm.

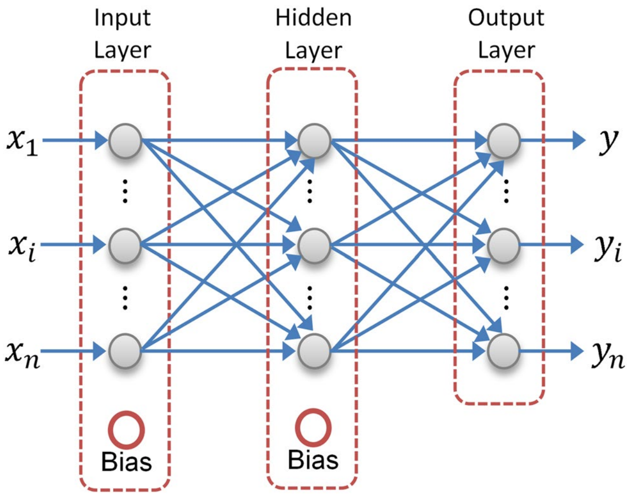

Fig. 2 A general three-layer FNN

4 Vortex search optimization algorithm

Vortex Search (VS) is a single-solution based algorithm for numerical function optimization, which has been recently proposed by [9]. VS algorithm has an inspiration originated from the vortex rotation shape of liquids. The algorithm narrows the search space along with iterations by using the inverse incomplete gamma function so that it reaches the global optimum faster. In addition, VS does not contain any user-tuning parameter.

Assuming a two-dimensional optimization problem, the schema of a vortex is formed by a series of nested circles. In the first iteration, the center of the largest circle $\left(\mu_0\right)$ is calculated by Eq. (3) for all dimensions.

$$\mu_0 = \frac{UpLimit + LowLimit}{2}, \tag{3}$$

where *UpLimit* and *LowLimit* are the upper and lower boundaries of decision variables, respectively. Also, the initial radius $\left(r_0\right)$ is determined by Eq. (4). Initially, a large value for r_0 is selected since the search area must be fully covered by the outer circle.

$$r_0 = \sigma_0 = \frac{max(UpLimit) - min(LowLimit)}{2}. \tag{4}$$

Then the candidate solutions are randomly generated by using Gaussian distribution within the specified circle in d-dimensional search space. The formulation of Gaussian distribution is given in Eq. (5).

$$p(x|\mu,\Sigma) = \frac{1}{\sqrt{(2\Pi)^d|\Sigma|}}exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \\ \Sigma = \sigma^2.[\mathrm{I}]_{dxd} \tag{5}$$

where x is the $dx1$ dimensional vector of a random variable and Σ is the covariance matrix. σ^2 shows the variance of the distribution. I indicates the dxd equivalent matrix. After the generation of candidate solutions, the parameter values of solutions outside the boundaries are drawn into the boundaries. This operation is shown in Eq. (6).

$$\mathrm{s}_m^j = \begin{cases} \mathrm{s}_m^j < LowLimit^j \\ \mathrm{rnd.}\left(UpLimit^j - LowLimit^j\right) + LowLimit^j \\ \\ LowLimit^j \precsim \mathrm{s}_m^j \precsim UpLimit^j \\ \mathrm{s}_m^j \\ \\ LowLimit^j,\ \mathrm{s}_m^j > UpLimit^j \\ \mathrm{rnd.}\left(UpLimit^j - LowLimit^j\right) + LowLimit^j \end{cases}, \tag{6}$$

where $m = 1,2,\ldots,n$ and $j = 1,2,\ldots,d$ and rnd is a random value in the range [0, 1].

The next operation is the selection of the best solution from the candidate solutions set. The best means that a solution has the best fitness value among all current solutions. A greedy selection is applied between the selected solution and the best solution found up to now. The better one is stored in memory and replaced with the current center of circle μ. In this phase, the circle radius is decreased so that exploitation increases in search space. This radius decrement is slower in the early iterations of the algorithm since exploration is prioritized. However, the reduction of the radius is accelerated to ensure better exploitation during the second half of the total iterations. In the VS algorithm, an adaptive radius reduction strategy is used with the incomplete inverse gamma function to reduce the radius. The gamma function $\Gamma(a)$ is given in Eq. (7).

$$\begin{aligned} &\Gamma(a) = \gamma(\mathrm{x},\mathrm{a}) + \Gamma(x,a) \\ &\gamma(\mathrm{x},\mathrm{a}) = \int_0^{\mathrm{x}} \mathrm{e}^{-\mathrm{t}}\mathrm{t}^{\mathrm{a}-1}\mathrm{dt}\ a > 0 \\ &\Gamma(x,a) = ?_x^{\alpha} e^{-t}t^{a-1}dta > 0 \end{aligned} \tag{7}$$

where $\gamma(\mathrm{x},\mathrm{a})$ and $\Gamma(x,a)$ are incomplete gamma function and complementary, respectively. a defines the inflexibility of the search, x is a random variable. In each iteration, the radius is updated by Eq. (8).

$$\begin{aligned} &r_t = \sigma_0.\frac{1}{x}.\Gamma\left(x,a_0\right) \\ &a_t = a_0 - t/MaxItr. \end{aligned} \tag{8}$$

Here, in the first iteration a_0 equals 1 in order to cover the entire search space. t refers to the iteration number and $MaxItr$ represents the maximum iteration number. This loop is repeated until a termination condition is satisfied.

5 VS-based feed-forward neural network

Several improvements can be made to increase the performance of FNNs. Some of the major ones are grouped into three headings. The first one is to compose the architecture of the FNN. In other words, changing the number of hidden layers and the number of neurons in each hidden layer is a crucial parameter to influence learning performance. The second heading is to determine the transfer function and optimal user-defined values (such as epoch size and learning rate) of the training method for the problem being addressed. Lastly, to determine the best connection weights is another heading.

In this study, we focus on the third heading, and the VS algorithm is used to determine the best weights and biases by minimizing the error in an FNN. It is executed as a training method by tuning all weights in FNN with a specified architecture. For this purpose, weights and biases represent decision variables of search space in the optimization process. These are defined in a matrix and candidate solutions are generated concerning that matrix.

Figure 3 indicates the distributions of the weights and biases for an FNN with a 2–2–1 structure. In this study, the matrix encoding strategy shown below is used as it is dealing with training FNNs. An encoded candidate solution matrix (CS) consists of four weight vectors, which are W1, B1, W2, and B2. These are represented as follows:

$$W1 = \begin{bmatrix} W_{13} & W_{23} \\ W_{14} & W_{24} \end{bmatrix}$$

$$B1 = \begin{bmatrix} W_{B1} \\ W_{B2} \end{bmatrix}$$

$$W2' = \begin{bmatrix} W_{36} \\ W_{46} \end{bmatrix}$$

$$B2 = \begin{bmatrix} W_{B3} \end{bmatrix}$$

$$CS = \left\{ W1\ B1\ \ W2'\ B2 \right\} = \{\mathrm{W13},\ \mathrm{W14},\ \mathrm{W23},\ \mathrm{W24},\ \mathrm{WB1},\ \mathrm{WB2},\ \mathrm{W36},\ \mathrm{W46},\ \mathrm{WB3}\}$$

where W1 is the hidden layer weight matrix, B1 is the hidden layer bias matrix, W2 is the output layer weight matrix, W2′ is the transpose of W2, and B2 is the hidden layer bias matrix.

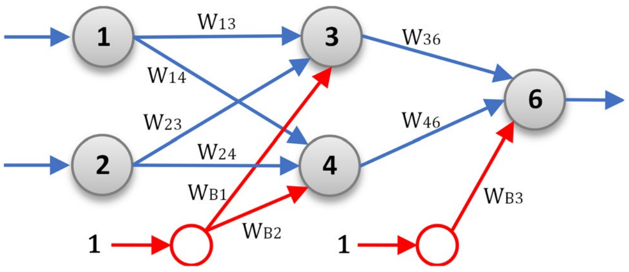

Fig. 3 A candidate solution for a 2–2–1 FNN

The training of FNN with VS, i.e. VS-FNN method starts to run after a suitable FNN architecture is defined for the selected dataset. To achieve the best values of weights and biases, the search space is constructed depending on the CS matrix. It means that the dimension of the problem equals the total number of weights and biases. Therefore, similar to how large-scale optimization problems are handled, the number of candidate solutions to be generated in one iteration in the complex FNN architectures can be increased compared to FNNs with a smaller dimension. The next step of the algorithm is to calculate the first center point and radius by using Eqs. (3) and (4), respectively. Then, the candidate solutions are randomly generated within the determined radius around this center. The solution with the least training error is selected as the new center point for the next iteration, and the search space is narrowed by decreasing the radius. Thus, the center point is evolved during the iterations.

Since the training of FNN is considered as an optimization problem, an objective function has to be defined to evaluate the fitness of the weight values to be generated during the optimization process. In this proposed approach, the mean training error is used as an objective function. Therefore, the feed-forward calculation is first applied for each sample in the dataset. Then, the errors which are the difference between calculated and expected values are found. Finally, the mean squared error (MSE) is calculated in Eq. (9).

$$\mathrm{MSE}\left(\vec{\omega}(t)\right) = \frac{1}{N}\sum_{|=1}^{N}\sum_{k=1}^{P}\left(d_k - o_k\right)^2, \tag{9}$$

where $\vec{\omega}(t)$ connection weights in iteration t. d_k represents the desired output value and o_k produced output value; P shows the number of output neurons and N is the number of samples.

VS-FNN continues until the maximum number of cycles is reached. The flowchart of VS-FNN can be seen in Fig. 4.

6 Experimental results

In this study, the proposed VS based learning method (VS-FNN) compares with PSO, ABC, GA, SA, and SGD algorithms by applying on an FNN with the same structure for all methods. These methods are called PSO-FNN, ABC-FNN, GA-FNN, SA-FNN, and SGD, respectively. To evaluate the performance of the proposed VS-FNN method, all methods are applied to six benchmark problems. The first of them is the three-bits parity (3-bit XOR) problem, whose inputs and expected outputs are shown in Table 1. The other problems are five popular datasets which are stated in the UCI machine learning repository of University of California at Irvine [10]. These are the Iris classification, Wine-Recognition, Wisconsin-Breast-Cancer, Pima-Indians-Diabetes, and

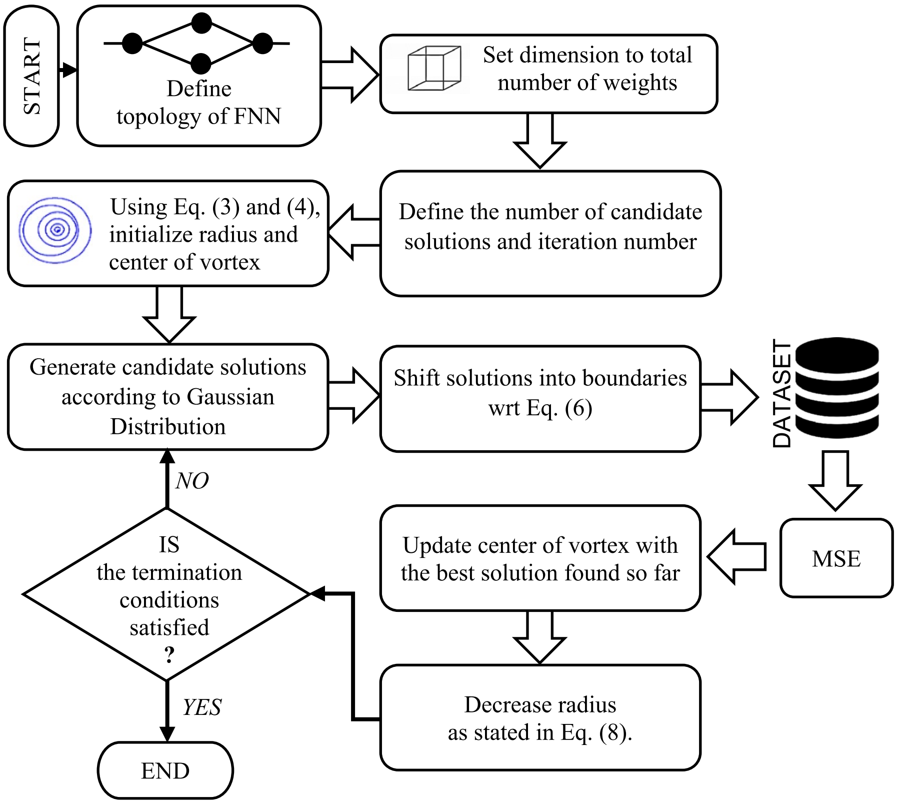

Fig. 4 Flowchart of VS-based learning method for FNN

Table 1 Three bits parity problem

Input	Output
000	0
001	1
010	1
011	0
100	1
101	0
110	0
111	1

Table 2 Benchmark datasets

Problem	Number of attributes	Number of classes	Number of Samples
3-bit parity	3	2	8
Iris	4	3	150
Wine	13	3	178
WBC	10	2	699
PID	8	2	768
Thyroid	21	3	7193

Thyroid-Disease. In this paper, for simplicity, the names of datasets are abbreviated in some places as iris, wine, WBC, PID, thyroid respectively.

These datasets are well-known benchmark problems and frequently used to evaluate for classification performance of the algorithms in the literature. The numbers of attributes, classes, and samples in each dataset are available in Table 2. In addition, attributes of datasets can be seen in Fig. 5.

To compare the results consistently, the same values are assigned to the common parameters in all algorithms. These are listed in Table 3. In these benchmark problems, it is assumed that a single hidden layer is used and every weight value is randomly initialized in the range of [− 50, 50]. The maximum number of iterations set to 100 for all problems, but only 500 for the three-bit parity problem.

In addition, there are individual user-tuning parameters for each algorithm. For the PSO algorithm, fully connected topology is used. The cognitive constant (C1) and social constant (C2) are set to 2. The parameters r_1 and r_2 are randomly generated in the range of [0,1]. *w* decreases linearly from 0.9 to 0.1. The initial velocities of particles are assigned to one-tenth of their initial *pbest* values.

For the ABC algorithm, the limit value is set to the product of the problem dimension and food number. For the GA, a real-coded algorithm is run over the problems. Binary tournament selection, arithmetic crossover (probability = 0.7),

Attr.	**Iris**	**Wine**	**PID**	**WBC**	**Thyroid**
x_1	sepal length	Alcohol	Pregnancies	radius	age
x_2	sepal width	Malic acid	Glucose	texture	sex
x_3	petal length	Ash	Blood Pressure	perimeter	on thyroxine
x_4	petal width	Alkalinity of ash	Skin Thickness	area	query on thyroxine
x_5		Magnesium	Insulin	smoothness	on antithyroid med.
x_6		Total phenols	BMI	compactness	sick
x_7		Flavonoids	Diabetes Pedigree Func.	concavity	pregnant
x_8		Nonflavonoid phenols	Age	concave points	thyroid surgery
x_9		Proanthocyanins	Outcome	symmetry	I131 treatment
x_{10}		Color intensity		fractal dimension	query hypothyroid
x_{11}		Hue			query hyperthyroid
x_{12}		OD280/OD315 of diluted			lithium
x_{13}		Proline			goiter
x_{14}					tumor
x_{15}					hypopituitary
x_{16}					psych
x_{17}					TSH
x_{18}					T3
x_{19}					TT4
x_{20}					T4U
x_{21}					FTI

Fig. 5 The list of attributes for each dataset

Table 3 Common parameters for all algorithms

Parameter	Value
Number of candidate solutions (population size)	50
Max. number of iterations	500 for 3-bit parity, 100 for the rest
Initial range of weights and biases	[− 50, 50]

and uniform mutation (probability = 0.01) operators are used. For the SA algorithm, the initial temperature is set to 1000 and the final temperature is 1. For the SGD algorithm, the learning rate is set to 0.2 and the number of epochs is 2000.

The outcomes of the methods are compared based on the average, standard deviation (Std dev), median, interquartile range (IQR), best and worst of the Mean Square Errors (MSE) values over 30 independent runs. Besides, the elapsed times for each run on all test problems are memorized and average times in second is calculated for each algorithm.

6.1 Selected FNN structures

In this study, the attributes of the datasets are normalized in the range of [0,1]. Then, all algorithms are applied to the datasets with the same architecture. In the selected FNN structures, two bias units are located into a network with one for the hidden layer and one for the output layer, whereas the sigmoid function is used as the activation function for both layers.

For each dataset, the FNNs with structure I–H–O are used to solve the classification problems, where I is the number of attributes, O is the number of classes, and H is the number of hidden nodes. The performances of FNNs are compared with H = 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 20, and 30. For these 12 different structures, each test problem is trained by the algorithms for 30 independent runs. The training phase continues until the maximum number of iterations has been completed.

6.2 Results and discussion

The statistical results and the comparisons of the ranks of the algorithms for each test problem are provided in tables from Tables 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 and 21. For any dataset, the best results are indicated in bold type in the tables. Further, the convergence curves are depicted in figures from Figs. 6, 7, 8, 9, 10 and 11. The parts (a), (b), (c), (d), (e), (f), (g), (h), (i), (j), (k), and (l) of each figure are the convergence curves for FNNs with H = 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 20, and 30, respectively.

Considering the experimental results, convergence graphics, and statistical results given in the tables; it is seen that the proposed VS-FNN method can mostly achieve more successful results in all data sets than all other algorithms compared. The main reason why VS-FNN can be superior to other algorithms can be summarized as having a more efficient exploitation process by iteratively narrowing the search space around the global optimum.

As known, the running principle of metaheuristics depends on two processes: exploration and exploitation. Therefore, the performance of an algorithm depends on how balanced it manages these two processes. In general, it is hoped that as the iterations progress, the algorithm convergences the global optimum depending on the exploration strategy. Then, for the exploitation process to be implemented effectively, new candidate solutions produced by neighborhood relationships begin to be close to each other in the search space.

At this point, VS employs a strong locality in the exploitation process, unlike many other algorithms. This means that VS allows more efficient exploitation by rapidly narrowing the search space systematically, especially after half of the maximum number of iterations. In this paper, it has been proven that VS-FNN has an ability to convergence to global optimum more effectively when it is applied with suitable control parameter values for a selected dataset. This inference can be seen from the convergence graphics given below. Especially after the 50th iteration, VS-FNN provides faster convergence than other algorithms.

Another advantage of VS-FNN that can be deduced from the experimental results is that the runtime is often less than other algorithms since the VS algorithm has a very simple structure in terms of computational complexity. In the VS algorithm, instead of generating new candidate solutions by mating with complex neighborhood relationships, it simply uses Gaussian distribution to generate the solutions in the narrowed search space. This simple strategy makes the algorithm run fast.

6.2.1 The N bit parity problem

The N bit parity problem has frequently been used to prove the effectiveness of training algorithms in several studies since it is considered as a difficult task for neural networks. It is defined as the number of even parity bits in a binary string. The XOR result of the bits in a string returns as output. FNNs with structures 3–H–1 are trained by the algorithms. Table 4 shows the statistical results of the 3-bit parity problem. In Table 4, the values below 10^{-20} are considered as zero. For all hidden nodes, it can be seen that VS-FNN and PSO-FNN have better results for especially median, best, mean, standard deviation, and IQR of MSE. The rank-based comparisons of the algorithms for the 3-bits

Table 4 The statistical results of MSE over 30 independent runs for the a 3-bit XOR problem

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
4	VS-FNN	2.81253E−02	4.10474E−02	**0.00000E+00**	8.33334E−02	**0.00000E+00**	9.37517E−02	1.16531E+00
	PSO-FNN	1.87500E−02	3.30990E−02	**0.00000E+00**	3.12500E−02	**0.00000E+00**	9.37500E−02	1.15099E+00
	ABC-FNN	**4.32147E−18**	**8.50082E−18**	1.17367E−18	**3.27015E−18**	**0.00000E+00**	**3.72820E−17**	3.01558E+00
	GA-FNN	2.38062E−02	4.27654E−02	1.17413E−08	3.22652E−02	**0.00000E+00**	1.26274E−01	2.81043E+00
	SA-FNN	4.21122E−01	1.12049E−01	4.99627E−01	1.24999E−01	1.25000E−01	5.00000E−01	3.83532E+00
	SGD	4.50321E−01	9.93797E−02	5.00000E−01	4.17943E−02	1.85831E−01	5.00031E−01	**2.07088E−01**
5	VS-FNN	8.68064E−03	2.65342E−02	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	9.37517E−02	1.19906E+00
	PSO-FNN	7.98611E−03	2.47236E−02	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	9.37500E−02	1.15974E+00
	ABC-FNN	**1.31334E−18**	**3.48205E−18**	2.69240E−19	6.28522E−19	**0.00000E+00**	**1.82585E−17**	2.88737E+00
	GA-FNN	1.28057E−02	2.73413E−02	4.29874E−10	6.96592E−06	**0.00000E+00**	9.37500E−02	2.75186E+00
	SA-FNN	4.29323E−01	8.71963E−02	4.58437E−01	1.25000E−01	2.50000E−01	5.83336E−01	3.82146E+00
	SGD	4.63183E−01	9.99852E−02	5.00000E−01	1.55741E−03	1.87595E−01	6.24997E−01	**1.87732E−01**
6	VS-FNN	5.20838E−03	2.02415E−02	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	9.37515E−02	1.40630E+00
	PSO-FNN	4.16667E−03	1.58568E−02	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	6.25000E−02	1.17447E+00
	ABC-FNN	**6.58967E−18**	**1.67519E−17**	1.87063E−19	1.73355E−18	**0.00000E+00**	**6.61549E−17**	2.92779E+00
	GA-FNN	1.03645E−02	2.33866E−02	8.46748E−12	1.06084E−04	**0.00000E+00**	6.55787E−02	2.75368E+00
	SA-FNN	4.36317E−01	1.10998E−01	5.00000E−01	1.25000E−01	1.95370E−01	6.25000E−01	3.85117E+00
	SGD	4.70340E−01	8.87540E−02	5.00000E−01	4.22504E−05	2.50037E−01	6.24998E−01	**1.66290E−01**
7	VS-FNN	2.05814E−03	1.12729E−02	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	6.17441E−02	1.41397E+00
	PSO-FNN	3.12500E−03	1.71163E−02	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	9.37500E−02	1.18591E+00
	ABC-FNN	**6.67647E−19**	**1.19500E−18**	7.08130E−20	1.03036E−18	**0.00000E+00**	**5.51390E−18**	2.91289E+00
	GA-FNN	9.45976E−03	3.19237E−02	3.01607E−14	2.15532E−11	**0.00000E+00**	1.25000E−01	2.78068E+00
	SA-FNN	4.47116E−01	1.22925E−01	5.00000E−01	1.25000E−01	1.95208E−01	7.49491E−01	3.91224E+00
	SGD	4.50071E−01	7.77059E−02	5.00000E−01	1.24995E−01	2.50006E−01	5.01791E−01	**1.81712E−01**
8	VS-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.37406E+00
	PSO-FNN	4.17280E−17	2.28553E−16	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.25184E−15	1.22662E+00
	ABC-FNN	5.20684E−19	1.00108E−18	9.72768E−20	5.88016E−19	**0.00000E+00**	4.74893E−18	2.93897E+00
	GA-FNN	1.45843E−02	3.91295E−02	5.30008E−15	9.32923E−10	**0.00000E+00**	1.25000E−01	2.74471E+00
	SA-FNN	4.89297E−01	4.78223E−02	5.00000E−01	1.60382E−08	3.46340E−01	6.25000E−01	3.97669E+00
	SGD	4.73604E−01	1.01737E−01	5.00000E−01	1.24945E−01	2.71662E−01	7.49992E−01	**1.68943E−01**
9	VS-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.31289E+00
	PSO-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.22828E+00
	ABC-FNN	5.12610E−19	1.24971E−18	5.41535E−20	5.76122E−19	**0.00000E+00**	6.57982E−18	2.94271E+00
	GA-FNN	2.16747E−03	8.43803E−03	9.39086E−17	2.25795E−12	**0.00000E+00**	3.92826E−02	2.76304E+00
	SA-FNN	4.43851E−01	1.11419E−01	5.00000E−01	1.25000E−01	1.35309E−01	6.24992E−01	3.92528E+00
	SGD	4.29189E−01	9.09436E−02	4.99999E−01	1.25000E−01	2.50011E−01	5.00002E−01	**1.83354E−01**
10	VS-FNN	5.35636E−17	2.93376E−16	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.60689E−15	1.32157E+00
	PSO-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.26948E+00
	ABC-FNN	8.11258E−19	3.66685E−18	**0.00000E+00**	1.63222E−19	**0.00000E+00**	2.01682E−17	2.63312E+00
	GA-FNN	4.16926E−03	2.28213E−02	**0.00000E+00**	1.07797E−15	**0.00000E+00**	1.25000E−01	2.79697E+00
	SA-FNN	4.40484E−01	9.75300E−02	4.99408E−01	1.25000E−01	2.50000E−01	6.25000E−01	3.97579E+00
	SGD	4.66620E−01	7.99221E−02	5.00000E−01	1.24910E−01	2.50013E−01	6.25001E−01	**1.79714E−01**
11	VS-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.33583E+00
	PSO-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.25067E+00
	ABC-FNN	4.12877E−19	1.16770E−18	**0.00000E+00**	4.27587E−20	**0.00000E+00**	5.06896E−18	2.62581E+00
	GA-FNN	8.33333E−03	3.17135E−02	**0.00000E+00**	3.02159E−15	**0.00000E+00**	1.25000E−01	2.79018E+00
	SA-FNN	4.49834E−01	9.87306E−02	4.99998E−01	1.25000E−01	2.50000E−01	6.24862E−01	4.12019E+00
	SGD	4.79172E−01	9.89235E−02	5.00000E−01	5.45780E−06	2.50037E−01	6.25006E−01	**1.83578E−01**

Table 4 (continued)

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
13	VS-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.37323E+00
	PSO-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.31777E+00
	ABC-FNN	3.04441E−19	6.46574E−19	2.65388E−20	2.78501E−19	**0.00000E+00**	3.13811E−18	2.62967E+00
	GA-FNN	4.16821E−03	2.28215E−02	**0.00000E+00**	1.04528E−15	**0.00000E+00**	1.25000E−01	2.78147E+00
	SA-FNN	4.60598E−01	8.18351E−02	5.00000E−01	1.06377E−01	2.49998E−01	6.25000E−01	4.13035E+00
	SGD	4.37679E−01	1.28998E−01	5.00000E−01	1.24998E−01	1.25052E−01	7.50027E−01	**1.82338E−01**
15	VS-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.38566E+00
	PSO-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.37541E+00
	ABC-FNN	8.22794E−19	2.07104E−18	4.57492E−20	2.52129E−19	**0.00000E+00**	8.58772E−18	2.66205E+00
	GA-FNN	1.66667E−02	4.32182E−02	**0.00000E+00**	7.49759E−14	**0.00000E+00**	1.25000E−01	2.82545E+00
	SA-FNN	4.37491E−01	9.66934E−02	4.99967E−01	1.24998E−01	2.49999E−01	6.24089E−01	4.09609E+00
	SGD	4.70844E−01	8.48553E−02	5.00000E−01	1.24933E−01	2.50001E−01	6.25001E−01	**2.00956E−01**
20	VS-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.42612E+00
	PSO-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.47335E+00
	ABC-FNN	1.05302E−18	3.27401E−18	4.62017E−20	5.54875E−19	**0.00000E+00**	1.75044E−17	2.69696E+00
	GA-FNN	4.90909E−18	2.68608E−17	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.47127E−16	2.82256E+00
	SA-FNN	4.96487E−01	7.98278E−02	5.00000E−01	1.12868E−02	3.75000E−01	7.50000E−01	4.20681E+00
	SGD	4.75004E−01	1.28816E−01	5.00000E−01	1.24978E−01	1.25030E−01	7.50000E−01	**1.99217E−01**
30	VS-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.51256E+00
	PSO-FNN	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.66350E+00
	ABC-FNN	1.03179E−18	2.06993E−18	4.41154E−20	8.95959E−19	**0.00000E+00**	7.54840E−18	2.81451E+00
	GA-FNN	1.25000E−02	3.81411E−02	**0.00000E+00**	**0.00000E+00**	**0.00000E+00**	1.25000E−01	2.91795E+00
	SA-FNN	5.13800E−01	1.02153E−01	5.00000E−01	9.13553E−08	3.74986E−01	7.50000E−01	4.60325E+00
	SGD	4.70834E−01	1.02148E−01	5.00000E−01	1.24994E−01	2.50000E−01	6.25002E−01	**2.04897E−01**

The best results are indicated in bold

parity problem are provided in Tables 5 and 6 according to the average and median of MSE, respectively. As shown in these tables, for all hidden nodes, the VS-FNN and PSO-FNN methods outperform all other methods by a majority. For FNNs with H = 4, 5, 6, and 7, the ABC-FNN method has better results for average MSE. These results prove that VS-FNN has a better or competitive ability to avoid local minima in this test problem. Figure 6 shows the convergence curves of all algorithms based on the average of MSE values. These figures confirm that VS-FNN reaches the most accurate results with a steady convergence acceleration.

Table 5 The comparison of the ranks of the algorithms for the results of 3-bit XOR problem with respect to average MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	7	7	3	1	0	0
PSO-FNN	7	9	2	0	0	0
ABC-FNN	4	2	6	0	0	0
GA-FNN	0	0	1	11	0	0
SA-FNN	0	0	0	0	7	5
SGD	0	0	0	0	5	7

Table 6 The comparison of the ranks of the algorithms for the results of 3-bit XOR problem with respect to median MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	12	12	6	2	0	0
PSO-FNN	12	12	6	2	0	0
ABC-FNN	2	2	8	6	0	0
GA-FNN	6	6	6	8	0	0
SA-FNN	0	0	0	0	11	3
SGD	0	0	0	0	3	11

Table 7 The statistical results of MSE over 30 independent runs for the Iris benchmark problem

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
4	VS-FNN	2.43644E−01	2.32396E−02	2.30762E−01	2.51331E−02	2.24072E−01	2.88072E−01	2.21769E+01
	PSO-FNN	2.43721E−01	2.57723E−02	**2.27714E−01**	3.15478E−02	2.18457E−01	2.88006E−01	2.51218E+01
	ABC-FNN	**2.43438E−01**	**1.43274E−02**	2.37932E−01	**1.96584E−02**	2.24505E−01	**2.81439E−01**	2.51235E+01
	GA-FNN	2.51232E−01	2.84178E−02	2.45282E−01	5.63393E−02	**2.07588E−01**	2.88335E−01	**1.62850E+01**
	SA-FNN	4.88103E−01	9.62118E−02	4.98700E−01	1.12008E−01	3.34726E−01	6.51042E−01	4.19788E+01
	SGD	4.50337E−01	9.85876E−02	4.68566E−01	1.41424E−01	2.77644E−01	6.48143E−01	1.99894E+01
5	VS-FNN	2.51251E−01	2.76028E−02	2.37649E−01	5.84197E−02	2.24967E−01	2.91127E−01	2.23816E+01
	PSO-FNN	**2.45570E−01**	2.78900E−02	**2.27862E−01**	5.43117E−02	2.19225E−01	2.99698E−01	2.69805E+01
	ABC-FNN	2.50683E−01	**1.95964E−02**	2.48251E−01	**3.19064E−02**	2.21962E−01	**2.87401E−01**	2.54329E+01
	GA-FNN	2.65331E−01	2.91281E−02	2.81347E−01	5.79564E−02	**2.17232E−01**	3.10534E−01	**2.04665E+01**
	SA-FNN	5.01198E−01	9.51846E−02	4.99597E−01	1.49226E−01	3.48958E−01	6.51042E−01	4.47891E+01
	SGD	4.28087E−01	1.05619E−01	4.12381E−01	1.61967E−01	2.84403E−01	6.51042E−01	2.06078E+01
6	VS-FNN	2.53644E−01	2.59992E−02	**2.39253E−01**	5.49045E−02	2.19577E−01	**2.89236E−01**	2.23785E+01
	PSO-FNN	**2.51625E−01**	2.57743E−02	2.48805E−01	5.45045E−02	**2.17465E−01**	2.89480E−01	2.64145E+01
	ABC-FNN	2.66707E−01	**1.86664E−02**	2.70954E−01	3.17181E−02	2.28049E−01	2.90711E−01	2.64205E+01
	GA-FNN	2.74379E−01	2.51352E−02	2.85257E−01	**2.19374E−02**	2.21772E−01	3.04955E−01	2.27940E+01
	SA-FNN	5.04667E−01	1.05356E−01	5.00000E−01	1.75136E−01	3.48958E−01	6.51042E−01	4.64830E+01
	SGD	4.42995E−01	1.01620E−01	4.47718E−01	1.49608E−01	3.05643E−01	6.51042E−01	**2.04396E+01**
7	VS-FNN	**2.56967E−01**	2.59907E−02	**2.43033E−01**	5.13651E−02	2.21537E−01	**2.93322E−01**	2.25201E+01
	PSO-FNN	2.60751E−01	2.78044E−02	2.60719E−01	5.64889E−02	2.22399E−01	3.14296E−01	2.61276E+01
	ABC-FNN	2.70521E−01	**1.87466E−02**	2.74445E−01	**3.03190E−02**	2.31289E−01	2.96760E−01	2.55828E+01
	GA-FNN	2.67934E−01	3.20405E−02	2.80677E−01	6.41614E−02	**2.08646E−01**	3.08057E−01	2.12566E+01
	SA-FNN	4.82616E−01	9.23159E−02	4.88392E−01	9.02903E−02	3.48911E−01	6.51032E−01	4.27638E+01
	SGD	4.18378E−01	9.84728E−02	3.64378E−01	1.54396E−01	2.92347E−01	6.51042E−01	**2.08107E+01**
8	VS-FNN	**2.59224E−01**	2.58611E−02	**2.54076E−01**	5.25551E−02	2.24390E−01	**2.92515E−01**	2.24496E+01
	PSO-FNN	2.64739E−01	2.46376E−02	2.75113E−01	4.62563E−02	2.25518E−01	2.96464E−01	2.57846E+01
	ABC-FNN	2.76062E−01	**1.75127E−02**	2.83838E−01	2.44351E−02	2.35940E−01	3.05245E−01	2.55501E+01
	GA-FNN	2.83008E−01	2.49745E−02	2.89209E−01	**1.59676E−02**	**2.22698E−01**	3.24944E−01	**1.99142E+01**
	SA-FNN	5.07529E−01	8.85660E−02	5.00000E−01	1.26454E−01	3.48958E−01	6.51042E−01	4.13954E+01
	SGD	4.60897E−01	9.93759E−02	4.64583E−01	1.22138E−01	3.17976E−01	6.51042E−01	2.05572E+01
9	VS-FNN	**2.69304E−01**	2.44796E−02	**2.81883E−01**	4.27004E−02	2.20452E−01	3.09808E−01	2.26455E+01
	PSO-FNN	2.73085E−01	2.36607E−02	2.83131E−01	3.38930E−02	2.21415E−01	**3.04808E−01**	2.57743E+01
	ABC-FNN	2.84420E−01	**1.77030E−02**	2.88154E−01	**1.91149E−02**	2.44274E−01	3.08507E−01	2.61580E+01
	GA-FNN	2.81710E−01	2.72651E−02	2.89409E−01	2.32139E−02	**2.09702E−01**	3.19141E−01	**2.02192E+01**
	SA-FNN	4.96000E−01	9.33461E−02	4.99956E−01	1.32147E−01	3.46670E−01	6.51042E−01	4.19177E+01
	SGD	4.47293E−01	9.39495E−02	4.73703E−01	1.49212E−01	2.85619E−01	6.51042E−01	2.09245E+01
10	VS-FNN	**2.71314E−01**	2.83168E−02	2.86203E−01	4.65293E−02	2.24963E−01	3.19211E−01	2.25233E+01
	PSO-FNN	2.77634E−01	2.34796E−02	2.86470E−01	1.23582E−02	**2.14016E−01**	**3.04976E−01**	2.61635E+01
	ABC-FNN	2.92109E−01	**9.13575E−03**	2.91514E−01	**1.00671E−02**	2.75061E−01	3.17941E−01	2.49879E+01
	GA-FNN	2.79553E−01	2.71543E−02	**2.85158E−01**	3.24312E−02	2.19490E−01	3.17895E−01	**2.00462E+01**
	SA-FNN	4.99258E−01	9.50989E−02	4.96681E−01	1.25762E−01	3.45897E−01	6.52339E−01	4.20646E+01
	SGD	4.67682E−01	1.01772E−01	4.70771E−01	1.36133E−01	3.13507E−01	6.51042E−01	2.08267E+01
11	VS-FNN	2.76234E−01	2.20663E−02	2.85660E−01	3.39275E−02	2.32582E−01	3.11502E−01	2.30217E+01
	PSO-FNN	**2.70703E−01**	2.37999E−02	**2.82350E−01**	3.44689E−02	**2.12419E−01**	**3.01821E−01**	2.61154E+01
	ABC-FNN	2.89406E−01	**1.58533E−02**	2.92388E−01	**1.99561E−02**	2.34628E−01	3.09443E−01	2.50827E+01
	GA-FNN	2.89250E−01	2.13633E−02	2.90545E−01	2.11975E−02	2.14641E−01	3.18298E−01	**2.01365E+01**
	SA-FNN	5.13702E−01	9.67552E−02	5.00934E−01	1.74390E−01	3.48958E−01	6.51041E−01	4.25447E+01
	SGD	4.70544E−01	1.08420E−01	4.94332E−01	1.60755E−01	3.07773E−01	6.51042E−01	2.12785E+01

Table 7 (continued)

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
13	VS-FNN	2.87599E−01	1.65396E−02	2.86388E−01	1.73680E−02	2.42275E−01	3.13805E−01	2.29639E+01
	PSO-FNN	**2.77697E−01**	2.52938E−02	**2.84818E−01**	1.88760E−02	**2.23850E−01**	**3.13035E−01**	2.65839E+01
	ABC-FNN	2.98348E−01	**1.12473E−02**	2.99336E−01	**1.46772E−02**	2.69705E−01	3.15825E−01	2.51252E+01
	GA-FNN	2.99240E−01	1.51098E−02	3.00569E−01	1.93312E−02	2.51644E−01	3.25551E−01	**2.04975E+01**
	SA-FNN	4.77607E−01	8.68518E−02	4.84968E−01	1.47201E−01	3.48958E−01	6.51042E−01	3.85769E+01
	SGD	4.41748E−01	8.40292E−02	4.56187E−01	1.36309E−01	3.14078E−01	6.50228E−01	2.12257E+01
15	VS-FNN	**2.85319E−01**	1.99811E−02	**2.92103E−01**	2.12015E−02	2.43082E−01	3.17321E−01	2.33243E+01
	PSO-FNN	2.90216E−01	1.68252E−02	2.93486E−01	1.80021E−02	**2.34899E−01**	3.16713E−01	2.68086E+01
	ABC-FNN	3.03838E−01	**6.52218E−03**	3.02802E−01	**7.89893E−03**	2.86648E−01	**3.16352E−01**	2.54751E+01
	GA-FNN	3.04385E−01	1.50171E−02	3.07721E−01	1.96697E−02	2.68207E−01	3.22954E−01	**1.89704E+01**
	SA-FNN	4.82326E−01	7.54325E−02	4.93157E−01	7.58386E−02	3.33442E−01	6.31137E−01	4.21316E+01
	SGD	4.40124E−01	9.67343E−02	4.07752E−01	1.62760E−01	3.10701E−01	6.30478E−01	2.16034E+01
20	VS-FNN	**2.93028E−01**	2.08771E−02	**2.97693E−01**	1.40496E−02	**2.28113E−01**	3.23420E−01	2.36517E+01
	PSO-FNN	3.01167E−01	1.32123E−02	3.05321E−01	1.83443E−02	2.62110E−01	**3.19861E−01**	2.82640E+01
	ABC-FNN	3.08349E−01	**1.11350E−02**	3.11786E−01	**1.33926E−02**	2.83328E−01	3.25524E−01	2.58536E+01
	GA-FNN	3.07650E−01	1.33853E−02	3.11082E−01	1.48770E−02	2.62784E−01	3.24970E−01	**1.85959E+01**
	SA-FNN	5.01532E−01	9.84196E−02	4.99037E−01	1.66368E−01	3.48958E−01	6.52765E−01	4.32749E+01
	SGD	4.53766E−01	8.85356E−02	4.76036E−01	1.25066E−01	3.30674E−01	6.40307E−01	2.15877E+01
30	VS-FNN	**3.05030E−01**	1.17588E−02	**3.04640E−01**	1.60361E−02	**2.74734E−01**	**3.21658E−01**	2.46304E+01
	PSO-FNN	3.08205E−01	1.27534E−02	3.14213E−01	1.59759E−02	2.82490E−01	3.29035E−01	2.90927E+01
	ABC-FNN	3.19329E−01	**8.76908E−03**	3.19901E−01	**1.10216E−02**	2.98348E−01	3.37702E−01	2.66697E+01
	GA-FNN	3.14991E−01	8.86047E−03	3.14732E−01	1.32024E−02	2.96195E−01	3.28493E−01	**1.93583E+01**
	SA-FNN	5.03735E−01	7.87360E−02	4.98245E−01	6.08416E−02	3.48958E−01	6.51693E−01	4.50483E+01
	SGD	4.71626E−01	1.02887E−01	4.74521E−01	1.56525E−01	3.34296E−01	6.49741E−01	2.32465E+01

The best results are indicated in bold

Table 8 The comparison of the ranks of the algorithms for the results of Iris benchmark problem with respect to average MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	7	4	1	0	0	0
PSO-FNN	4	7	1	0	0	0
ABC-FNN	1	1	4	6	0	0
GA-FNN	0	0	6	6	0	0
SA-FNN	0	0	0	0	0	12
SGD	0	0	0	0	12	0

Table 9 The comparison of the ranks of the algorithms for the results of Iris benchmark problem with respect to median MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	7	5	0	0	0	0
PSO-FNN	4	7	1	0	0	0
ABC-FNN	0	0	8	4	0	0
GA-FNN	1	0	3	8	0	0
SA-FNN	0	0	0	0	0	12
SGD	0	0	0	0	12	0

6.2.2 The iris classification problem

The iris classification problem is the most known dataset in the field of machine learning. This dataset contains the samples to classify the iris flowers into three species (Setosa, Versicolor, and Virginica) by the aid of the measurements of length and width of sepals and petals. FNNs with structures 4–H–3 are trained by the algorithms. The statistical results for the Iris benchmark problem are presented in Table 7. For average MSE values, VS-FNN has the best values in 7 of 12 FNNs with H = 7, 8, 9, 10, 15, 20, and 30. Moreover, the proposed method has the second rank in 4 of 12 FNNs with H = 4, 6, 11, and 13, while it has the third rank for only FNN with 5 hidden nodes among the six algorithms applied to this problem. Similarly, for median MSE values, the VS-FNN method has the best values in 7 of 12 FNNs, while it has the second-best values in the rest of the structures. The rank-based comparisons are given in Tables 8 and 9, regarding the mean and median of MSE values, respectively. As seen in these tables, it can be said that the winner method is the VS-FNN. Then PSO-FNN closely follows it. For the Iris problem, Fig. 7 shows the convergence curves of all algorithms based on the average of MSE values. It can be

Table 10 The statistical results of MSE over 30 independent runs for the Wine benchmark problem

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
4	VS-FNN	1.78052E−01	2.77159E−02	1.71403E−01	4.24807E−02	1.35395E−01	2.43677E−01	5.08390E+00
	PSO-FNN	**1.68631E−01**	3.24623E−02	**1.66823E−01**	**4.18680E−02**	**1.20329E−01**	**2.39664E−01**	4.99960E+00
	ABC-FNN	2.30296E−01	**2.57588E−02**	2.26380E−01	4.59484E−02	1.73919E−01	2.75249E−01	5.19828E+00
	GA-FNN	1.96828E−01	4.26802E−02	1.84979E−01	8.54610E−02	1.34520E−01	2.62172E−01	4.52074E+00
	SA-FNN	4.71889E−01	9.39481E−02	4.86891E−01	1.30263E−01	2.88460E−01	6.66628E−01	8.51798E+00
	SGD	4.40430E−01	9.68909E−02	4.04567E−01	9.23331E−02	2.96025E−01	6.66667E−01	**3.72921E+00**
5	VS-FNN	1.81941E−01	3.81221E−02	1.81013E−01	**2.39939E−02**	**6.25440E−02**	**2.45968E−01**	5.07215E+00
	PSO-FNN	**1.80755E−01**	4.28604E−02	**1.69153E−01**	7.80437E−02	1.17242E−01	2.56060E−01	5.04113E+00
	ABC-FNN	2.29195E−01	**2.87319E−02**	2.26866E−01	3.37856E−02	1.74726E−01	2.83775E−01	5.21385E+00
	GA-FNN	1.81258E−01	3.91704E−02	1.76982E−01	6.30207E−02	1.25060E−01	2.56296E−01	4.52405E+00
	SA-FNN	4.87406E−01	9.91036E−02	4.93310E−01	1.01840E−01	2.43445E−01	6.66667E−01	8.58200E+00
	SGD	4.85096E−01	1.06277E−01	4.70542E−01	1.51017E−01	2.80807E−01	6.66667E−01	**3.64854E+00**
6	VS-FNN	**1.79783E−01**	**2.47650E−02**	1.75602E−01	**2.02920E−02**	1.34104E−01	**2.39664E−01**	5.23948E+00
	PSO-FNN	1.84182E−01	3.90338E−02	**1.74457E−01**	5.87134E−02	**1.18521E−01**	2.43677E−01	5.11154E+00
	ABC-FNN	2.22656E−01	3.08097E−02	2.19450E−01	4.84608E−02	1.63437E−01	2.78488E−01	5.17123E+00
	GA-FNN	1.84700E−01	3.15975E−02	1.78633E−01	4.23298E−02	1.24897E−01	2.40813E−01	4.58658E+00
	SA-FNN	5.25603E−01	8.07256E−02	5.13109E−01	1.12703E−01	3.92557E−01	6.66667E−01	8.66408E+00
	SGD	4.42216E−01	1.09021E−01	4.45693E−01	2.20974E−01	2.61100E−01	5.99251E−01	**3.73295E+00**
7	VS-FNN	1.83566E−01	**2.68452E−02**	1.75550E−01	3.41962E−02	1.40185E−01	2.57642E−01	5.06302E+00
	PSO-FNN	**1.71561E−01**	3.20103E−02	1.66627E−01	**1.74955E−02**	**1.14882E−01**	**2.31996E−01**	5.18409E+00
	ABC-FNN	2.39082E−01	3.14690E−02	2.35805E−01	4.53281E−02	1.88053E−01	3.00319E−01	5.26060E+00
	GA-FNN	1.76690E−01	3.83557E−02	**1.66389E−01**	3.21401E−02	1.18385E−01	2.62311E−01	5.03579E+00
	SA-FNN	4.64103E−01	9.79285E−02	4.86888E−01	1.53558E−01	2.30556E−01	5.99251E−01	8.61272E+00
	SGD	4.84277E−01	1.07538E−01	4.85019E−01	1.60096E−01	2.61100E−01	6.66667E−01	**3.81607E+00**
8	VS-FNN	1.75012E−01	2.39687E−02	1.69864E−01	1.96419E−02	1.31524E−01	2.37653E−01	5.08693E+00
	PSO-FNN	**1.61269E−01**	**1.69173E−02**	**1.60901E−01**	**9.68633E−03**	1.32242E−01	**2.13864E−01**	5.14796E+00
	ABC-FNN	2.32141E−01	3.49776E−02	2.41055E−01	5.33918E−02	1.44342E−01	2.95314E−01	5.22407E+00
	GA-FNN	1.77367E−01	3.47922E−02	1.67298E−01	3.32750E−02	**1.24059E−01**	2.59753E−01	5.15489E+00
	SA-FNN	5.06880E−01	9.40223E−02	4.99987E−01	1.50930E−01	3.09094E−01	6.66667E−01	8.77023E+00
	SGD	4.58552E−01	1.12823E−01	4.35991E−01	1.35455E−01	2.78742E−01	6.66667E−01	**3.74295E+00**
9	VS-FNN	**1.77830E−01**	2.84410E−02	1.71131E−01	**1.73075E−02**	1.32533E−01	2.56060E−01	5.17265E+00
	PSO-FNN	1.80078E−01	3.15166E−02	**1.70430E−01**	3.75483E−02	**1.25927E−01**	**2.38515E−01**	5.18595E+00
	ABC-FNN	2.53268E−01	**2.08966E−02**	2.56653E−01	2.59969E−02	1.96140E−01	2.85105E−01	5.28453E+00
	GA-FNN	1.93115E−01	3.91584E−02	1.80493E−01	7.57996E−02	1.36902E−01	2.67380E−01	6.47034E+00
	SA-FNN	4.97907E−01	1.12984E−01	5.10632E−01	2.01002E−01	3.22094E−01	6.70412E−01	8.81002E+00
	SGD	4.58672E−01	1.13081E−01	4.62365E−01	2.20974E−01	2.92637E−01	6.66667E−01	**3.73860E+00**
10	VS-FNN	1.82211E−01	2.93905E−02	1.70608E−01	4.36727E−02	**1.31074E−01**	**2.40973E−01**	5.13744E+00
	PSO-FNN	**1.74696E−01**	3.13578E−02	1.67025E−01	3.45438E−02	1.34035E−01	2.56060E−01	5.21630E+00
	ABC-FNN	2.42158E−01	**2.90449E−02**	2.48626E−01	4.74622E−02	1.82017E−01	2.86288E−01	5.37937E+00
	GA-FNN	1.82010E−01	3.47606E−02	**1.66205E−01**	**1.71427E−02**	1.43117E−01	2.54606E−01	6.09510E+00
	SA-FNN	4.97058E−01	1.06896E−01	5.08078E−01	1.13651E−01	3.16941E−01	6.66667E−01	8.76658E+00
	SGD	5.34982E−01	8.74598E−02	5.54307E−01	1.12360E−01	3.34698E−01	6.66667E−01	**3.85468E+00**
11	VS-FNN	1.90873E−01	3.22471E−02	1.75808E−01	4.82135E−02	1.51087E−01	2.56137E−01	5.13176E+00
	PSO-FNN	**1.73032E−01**	**2.64189E−02**	**1.68329E−01**	**1.80929E−02**	**1.40473E−01**	**2.41020E−01**	5.29481E+00
	ABC-FNN	2.45637E−01	2.98516E−02	2.48257E−01	3.28622E−02	1.68884E−01	3.04559E−01	5.31470E+00
	GA-FNN	2.00905E−01	3.40762E−02	1.93726E−01	7.14047E−02	1.56378E−01	2.55619E−01	6.59421E+00
	SA-FNN	5.14260E−01	9.27558E−02	5.20951E−01	9.53655E−02	3.13476E−01	6.66653E−01	8.80969E+00
	SGD	4.65455E−01	1.02844E−01	4.45693E−01	1.51053E−01	2.99838E−01	6.66667E−01	**3.88171E+00**

Table 10 (continued)

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
13	VS-FNN	1.89009E−01	3.63125E−02	1.75777E−01	5.74186E−02	**1.20384E−01**	2.48077E−01	5.20294E+00
	PSO-FNN	**1.72873E−01**	**2.64687E−02**	**1.68611E−01**	**8.55568E−03**	1.23657E−01	**2.40379E−01**	5.27239E+00
	ABC-FNN	2.43322E−01	2.76595E−02	2.47462E−01	3.30768E−02	1.60181E−01	2.94008E−01	5.35470E+00
	GA-FNN	1.82909E−01	3.45680E−02	1.69432E−01	4.35607E−02	1.32530E−01	2.65047E−01	6.34490E+00
	SA-FNN	4.74399E−01	1.19580E−01	4.81861E−01	2.20974E−01	2.92090E−01	7.07865E−01	8.89570E+00
	SGD	4.70354E−01	1.01603E−01	4.84198E−01	1.43864E−01	2.92805E−01	6.66667E−01	**3.88120E+00**
15	VS-FNN	1.83279E−01	3.22970E−02	1.72942E−01	4.32748E−02	1.25303E−01	2.45705E−01	5.28605E+00
	PSO-FNN	**1.69141E−01**	**2.81893E−02**	**1.65206E−01**	**1.50642E−02**	**1.13834E−01**	**2.41141E−01**	5.38307E+00
	ABC-FNN	2.48999E−01	2.99659E−02	2.53026E−01	3.81146E−02	1.80446E−01	2.96963E−01	5.40583E+00
	GA-FNN	1.87902E−01	3.20110E−02	1.74367E−01	3.76116E−02	1.47221E−01	2.60585E−01	6.58522E+00
	SA-FNN	5.07683E−01	1.00732E−01	4.86891E−01	1.52490E−01	3.33333E−01	6.66667E−01	9.03878E+00
	SGD	4.59631E−01	9.06534E−02	4.64731E−01	1.12360E−01	2.97581E−01	6.66667E−01	**3.93878E+00**
20	VS-FNN	1.78944E−01	**2.77788E−02**	1.75056E−01	2.65903E−02	1.23056E−01	**2.42212E−01**	5.41782E+00
	PSO-FNN	**1.72245E−01**	2.79370E−02	**1.64340E−01**	**1.29620E−02**	1.32663E−01	2.57390E−01	5.55475E+00
	ABC-FNN	2.52851E−01	3.66318E−02	2.61144E−01	5.46676E−02	1.58968E−01	3.06246E−01	5.51162E+00
	GA-FNN	1.83325E−01	3.33868E−02	1.71448E−01	3.20618E−02	**1.15303E−01**	2.48241E−01	6.52171E+00
	SA-FNN	4.91013E−01	1.22953E−01	5.06850E−01	1.89139E−01	2.66792E−01	6.85382E−01	9.17694E+00
	SGD	4.85418E−01	1.01896E−01	4.89383E−01	1.53558E−01	2.76365E−01	6.66667E−01	**3.99485E+00**
30	VS-FNN	1.87320E−01	3.26626E−02	1.70946E−01	5.48939E−02	1.44397E−01	2.41157E−01	5.71161E+00
	PSO-FNN	**1.68504E−01**	**1.87748E−02**	**1.64141E−01**	**1.05278E−02**	**1.29673E−01**	**2.39941E−01**	5.96127E+00
	ABC-FNN	2.62426E−01	2.68223E−02	2.62935E−01	3.09776E−02	2.04912E−01	3.19766E−01	5.73067E+00
	GA-FNN	2.10176E−01	3.60105E−02	2.21409E−01	6.59859E−02	1.40057E−01	2.59761E−01	6.84317E+00
	SA-FNN	5.20166E−01	1.10701E−01	5.13108E−01	1.53558E−01	3.32098E−01	7.35955E−01	9.70018E+00
	SGD	4.81746E−01	1.10639E−01	4.85019E−01	2.22593E−01	3.16479E−01	6.66667E−01	**4.30074E+00**

The best results are indicated in bold

Table 11 The comparison of the ranks of the algorithms for the results of Wine benchmark problem with respect to average MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	2	6	4	0	0	0
PSO-FNN	10	2	0	0	0	0
ABC-FNN	0	0	0	12	0	0
GA-FNN	0	4	8	0	0	0
SA-FNN	0	0	0	0	2	10
SGD	0	0	0	0	10	2

Table 12 The comparison of the ranks of the algorithms for the results of Wine benchmark problem with respect to median MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	0	6	6	0	0	0
PSO-FNN	10	2	0	0	0	0
ABC-FNN	0	0	0	12	0	0
GA-FNN	2	4	6	0	0	0
SA-FNN	0	0	0	0	2	10
SGD	0	0	0	0	10	2

Table 13 The statistical results of MSE over 30 independent runs for the WBC benchmark problem

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
4	VS-FNN	**2.27887E−01**	**1.08515E−02**	**2.25906E−01**	**1.08247E−15**	**2.25906E−01**	2.85342E−01	**1.88312E+01**
	PSO-FNN	2.29092E−01	1.12396E−02	2.25906E−01	3.43174E−09	**2.25906E−01**	2.85342E−01	2.26603E+01
	ABC-FNN	2.44396E−01	1.54253E−02	2.39396E−01	1.47726E−02	2.26026E−01	**2.84454E−01**	1.98827E+01
	GA-FNN	2.38208E−01	2.15081E−02	2.25907E−01	1.19859E−02	**2.25906E−01**	2.85346E−01	2.66458E+01
	SA-FNN	5.03800E−01	9.74800E−02	5.00000E−01	2.15650E−05	3.14838E−01	6.55222E−01	4.11185E+01
	SGD	4.48282E−01	1.31730E−01	4.48459E−01	1.55222E−01	2.41640E−01	6.55222E−01	1.92415E+01
5	VS-FNN	**2.25906E−01**	**1.11877E−15**	**2.25906E−01**	**1.02696E−15**	**2.25906E−01**	**2.25906E−01**	1.88813E+01
	PSO-FNN	2.29929E−01	1.27459E−02	2.25906E−01	4.19151E−09	**2.25906E−01**	2.85342E−01	2.28235E+01
	ABC-FNN	2.39949E−01	1.40523E−02	2.34220E−01	1.40589E−02	2.26117E−01	2.85342E−01	2.00345E+01
	GA-FNN	2.47485E−01	2.72103E−02	2.27373E−01	5.88877E−02	**2.25906E−01**	2.85362E−01	2.63961E+01
	SA-FNN	4.84540E−01	1.22279E−01	5.00000E−01	1.55222E−01	2.28576E−01	6.55222E−01	4.23792E+01
	SGD	4.49944E−01	1.44717E−01	4.74215E−01	2.06763E−01	2.41145E−01	6.55222E−01	**1.82221E+01**
6	VS-FNN	2.27887E−01	1.08515E−02	**2.25906E−01**	**1.97065E−15**	**2.25906E−01**	2.85342E−01	1.89524E+01
	PSO-FNN	**2.26334E−01**	**2.19645E−03**	2.25906E−01	6.75717E−08	**2.25906E−01**	**2.37953E−01**	2.39015E+01
	ABC-FNN	2.42684E−01	1.47269E−02	2.41387E−01	1.89753E−02	2.26766E−01	2.85348E−01	2.01618E+01
	GA-FNN	2.52129E−01	2.68215E−02	2.42855E−01	5.76716E−02	**2.25906E−01**	2.85457E−01	2.64250E+01
	SA-FNN	4.94891E−01	1.32108E−01	4.99732E−01	3.10443E−01	2.98734E−01	6.55222E−01	4.25728E+01
	SGD	4.72462E−01	1.21283E−01	4.99998E−01	5.18166E−02	2.41697E−01	6.55222E−01	**1.79393E+01**
7	VS-FNN	**2.26935E−01**	2.25754E−02	**2.25906E−01**	**1.44329E−15**	**1.37893E−01**	**2.85342E−01**	1.92852E+01
	PSO-FNN	2.30707E−01	**1.51595E−02**	2.25906E−01	3.36686E−05	2.25906E−01	2.85342E−01	2.37545E+01
	ABC-FNN	2.49979E−01	1.67419E−02	2.45841E−01	2.36330E−02	2.27966E−01	2.85405E−01	2.02216E+01
	GA-FNN	2.45131E−01	2.50809E−02	2.28154E−01	4.61001E−02	2.25906E−01	2.85863E−01	2.64440E+01
	SA-FNN	4.95716E−01	1.02577E−01	5.00000E−01	2.38787E−02	3.08098E−01	6.55222E−01	4.19494E+01
	SGD	4.73991E−01	1.47499E−01	5.00000E−01	3.10443E−01	2.41640E−01	6.55222E−01	**1.80196E+01**
8	VS-FNN	2.32548E−01	3.42378E−02	**2.25906E−01**	**3.77476E−15**	**9.67536E−02**	**2.85342E−01**	1.92162E+01
	PSO-FNN	**2.30426E−01**	**1.52269E−02**	2.25906E−01	1.94837E−06	2.25906E−01	2.85342E−01	2.26596E+01
	ABC-FNN	2.55432E−01	2.11951E−02	2.53438E−01	3.81779E−02	2.26037E−01	2.88578E−01	2.05241E+01
	GA-FNN	2.41019E−01	2.25925E−02	2.26473E−01	2.67536E−02	2.25906E−01	2.85434E−01	2.66813E+01
	SA-FNN	5.21737E−01	1.19858E−01	5.00000E−01	1.56453E−01	2.86107E−01	6.55222E−01	4.11480E+01
	SGD	4.54939E−01	1.12879E−01	4.74092E−01	1.55222E−01	2.41145E−01	6.55222E−01	**1.77715E+01**
9	VS-FNN	2.39775E−01	2.55684E−02	**2.25906E−01**	**1.66187E−12**	**2.25906E−01**	**2.85342E−01**	1.91639E+01
	PSO-FNN	**2.36370E−01**	2.23375E−02	2.25907E−01	2.42364E−03	**2.25906E−01**	2.85342E−01	2.28355E+01
	ABC-FNN	2.50776E−01	**2.00932E−02**	2.48260E−01	3.23271E−02	2.27933E−01	2.87280E−01	2.03303E+01
	GA-FNN	2.51461E−01	2.62786E−02	2.40156E−01	5.87945E−02	**2.25906E−01**	2.85432E−01	2.67783E+01
	SA-FNN	4.82405E−01	1.13542E−01	5.00000E−01	1.55222E−01	2.85332E−01	6.55222E−01	4.17043E+01
	SGD	4.42792E−01	1.22850E−01	4.74230E−01	1.55222E−01	2.40143E−01	6.55222E−01	**1.80979E+01**
10	VS-FNN	**2.29869E−01**	**1.50794E−02**	**2.25906E−01**	**2.91434E−15**	2.25906E−01	**2.85342E−01**	1.91876E+01
	PSO-FNN	2.32626E−01	2.16300E−02	2.25906E−01	5.98485E−05	**2.00099E−01**	2.85342E−01	2.31323E+01
	ABC-FNN	2.54580E−01	2.14700E−02	2.55064E−01	4.34408E−02	2.27124E−01	2.88132E−01	2.05010E+01
	GA-FNN	2.54967E−01	2.50194E−02	2.46963E−01	5.41523E−02	2.25906E−01	2.87160E−01	2.69544E+01
	SA-FNN	5.15047E−01	1.02700E−01	5.00000E−01	1.53387E−01	3.44743E−01	6.55222E−01	4.28006E+01
	SGD	4.51444E−01	1.21798E−01	5.00000E−01	1.55222E−01	2.38824E−01	6.55222E−01	**1.81410E+01**
11	VS-FNN	**2.29869E−01**	**1.50794E−02**	**2.25906E−01**	**1.55431E−15**	**2.25906E−01**	**2.85342E−01**	1.94313E+01
	PSO-FNN	2.36074E−01	2.24247E−02	2.25917E−01	6.43458E−04	2.25906E−01	2.85342E−01	2.38783E+01
	ABC-FNN	2.60055E−01	2.06561E−02	2.57350E−01	4.25267E−02	2.26079E−01	2.85855E−01	2.05511E+01
	GA-FNN	2.51931E−01	2.48726E−02	2.43998E−01	5.75371E−02	2.25906E−01	2.85799E−01	2.68990E+01
	SA-FNN	4.68567E−01	1.07421E−01	5.00000E−01	1.55222E−01	2.85342E−01	6.55222E−01	4.19529E+01
	SGD	5.20540E−01	1.20725E−01	5.00000E−01	2.06763E−01	2.89396E−01	6.55222E−01	**1.82800E+01**

Table 13 (continued)

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
13	VS-FNN	2.43737E−01	2.77026E−02	**2.25906E−01**	5.94360E−02	**2.25906E−01**	**2.85342E−01**	2.03337E+01
	PSO-FNN	**2.42990E−01**	2.65271E−02	2.25935E−01	5.94359E−02	**2.25906E−01**	2.85342E−01	2.40405E+01
	ABC-FNN	2.65107E−01	**2.02586E−02**	2.68195E−01	**3.34063E−02**	2.29053E−01	2.87044E−01	2.06707E+01
	GA-FNN	2.55779E−01	2.57491E−02	2.61156E−01	5.61671E−02	2.25906E−01	2.85842E−01	2.70442E+01
	SA-FNN	4.89070E−01	1.07280E−01	5.00000E−01	1.55222E−01	3.44767E−01	6.55222E−01	4.09423E+01
	SGD	4.46005E−01	1.14753E−01	5.00000E−01	1.55222E−01	2.37302E−01	6.55222E−01	**1.85069E+01**
15	VS-FNN	**2.34898E−01**	2.34817E−02	**2.25906E−01**	**3.24740E−15**	**1.98473E−01**	**2.85342E−01**	2.01217E+01
	PSO-FNN	2.47140E−01	2.78305E−02	2.26795E−01	5.94317E−02	2.25906E−01	2.85344E−01	2.42436E+01
	ABC-FNN	2.66701E−01	**2.28012E−02**	2.70912E−01	4.28868E−02	2.26372E−01	3.06675E−01	2.10119E+01
	GA-FNN	2.65616E−01	2.54044E−02	2.85051E−01	4.71488E−02	2.12557E−01	2.85864E−01	2.75930E+01
	SA-FNN	5.21117E−01	1.00517E−01	5.00000E−01	1.56425E−01	3.34045E−01	6.55222E−01	4.09297E+01
	SGD	4.66399E−01	9.90758E−02	5.00000E−01	5.53822E−02	2.37302E−01	6.55222E−01	**1.86668E+01**
20	VS-FNN	2.53643E−01	3.01588E−02	**2.25906E−01**	5.94360E−02	2.25906E−01	**2.85342E−01**	2.04646E+01
	PSO-FNN	**2.44593E−01**	3.74596E−02	2.26476E−01	5.93863E−02	**1.11514E−01**	2.85343E−01	2.46317E+01
	ABC-FNN	2.65418E−01	2.80002E−02	2.63937E−01	**4.92932E−02**	2.26485E−01	3.26511E−01	2.14614E+01
	GA-FNN	2.63498E−01	**2.65807E−02**	2.85339E−01	5.67209E−02	2.25910E−01	2.85941E−01	2.58147E+01
	SA-FNN	4.93234E−01	1.13622E−01	5.00000E−01	1.55222E−01	3.01855E−01	6.55222E−01	4.37537E+01
	SGD	5.04835E−01	1.30446E−01	5.00000E−01	3.10443E−01	2.87073E−01	6.55222E−01	**1.87079E+01**
30	VS-FNN	2.53643E−01	3.01588E−02	**2.25906E−01**	5.94360E−02	**2.25906E−01**	**2.85342E−01**	2.12991E+01
	PSO-FNN	**2.53030E−01**	2.66914E−02	2.35640E−01	5.81792E−02	2.25906E−01	2.85343E−01	2.55887E+01
	ABC-FNN	2.70776E−01	**2.31632E−02**	2.83846E−01	**2.48117E−02**	2.26734E−01	2.96900E−01	2.20557E+01
	GA-FNN	2.67763E−01	2.63110E−02	2.85342E−01	3.99872E−02	2.25906E−01	2.95394E−01	**1.89227E+01**
	SA-FNN	4.99999E−01	1.15295E−01	5.00000E−01	3.10421E−01	3.44778E−01	6.55222E−01	4.66993E+01
	SGD	4.83830E−01	1.00250E−01	5.00000E−01	5.93485E−02	3.44778E−01	6.55222E−01	1.94926E+01

The best results are indicated in bold

Table 14 The comparison of the ranks of the algorithms for the results of WBC benchmark problem with respect to average MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	6	6	0	0	0	0
PSO-FNN	6	6	0	0	0	0
ABC-FNN	0	0	4	8	0	0
GA-FNN	0	0	8	4	0	0
SA-FNN	0	0	0	0	2	10
SGD	0	0	0	0	10	2

Table 15 The comparison of the ranks of the algorithms for the results of WBC benchmark problem with respect to median MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	12	0	0	0	0	0
PSO-FNN	0	12	0	0	0	0
ABC-FNN	0	0	4	8	0	0
GA-FNN	0	0	8	4	0	0
SA-FNN	0	0	0	0	6	7
SGD	0	0	0	0	7	6

seen from these figures that, especially after the half of the maximum number of iterations, VS provides a faster convergence thanks to its strong locality feature.

6.2.3 The wine recognition problem

The wine recognition problem is a well-known classification dataset that contains the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. In this section, FNNs with structures 13–H–3 are trained by the algorithms. The comparative training results are shown in Table 10 for this problem. From Table 10, it can be seen that the VS-FNN method has the best average MSE values in only two FNNs with H = 6 and 9. Although the VS-FNN is the second-best values for the remaining FNN structures, it follows behind the PSO for this problem. However, the VS-FNN outperforms the ABC-FNN, GA-FNN, SA-FNN, and SGD methods by obtaining competitive results with PSO-FNN. The rank-based comparisons provided in Tables 11 and 12 confirm this situation. On the other hand, although the GA-FNN method can also achieve good results for this problem, it stays behind PSO, VS,

Table 16 The statistical results of MSE over 30 independent runs for the PID benchmark problem

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
4	VS-FNN	**2.36507E−01**	2.03596E−02	2.28426E−01	**4.80345E−03**	2.23884E−01	2.89236E−01	1.95710E+01
	PSO-FNN	2.39451E−01	2.33079E−02	**2.27876E−01**	1.26871E−02	2.14950E−01	2.87420E−01	1.97616E+01
	ABC-FNN	2.47195E−01	**1.43297E−02**	2.42180E−01	1.72198E−02	2.25979E−01	**2.79139E−01**	2.06209E+01
	GA-FNN	2.51232E−01	2.84178E−02	2.45282E−01	5.63393E−02	**2.07588E−01**	2.88335E−01	2.95836E+01
	SA-FNN	4.84378E−01	1.05575E−01	4.87785E−01	1.52207E−01	2.93109E−01	6.53646E−01	3.33734E+01
	SGD	4.58673E−01	1.10643E−01	4.55195E−01	1.30333E−01	2.88047E−01	6.51042E−01	**1.53193E+01**
5	VS-FNN	2.50893E−01	2.81529E−02	**2.31670E−01**	5.80468E−02	2.22029E−01	2.88959E−01	1.98153E+01
	PSO-FNN	**2.43302E−01**	2.62566E−02	2.33234E−01	4.92575E−02	**2.09338E−01**	**2.87130E−01**	2.07999E+01
	ABC-FNN	2.49702E−01	**2.01972E−02**	2.43713E−01	**3.40919E−02**	2.23390E−01	2.89178E−01	2.05967E+01
	GA-FNN	2.65331E−01	2.91281E−02	2.81347E−01	5.79564E−02	2.17232E−01	3.10534E−01	2.94261E+01
	SA-FNN	5.05740E−01	9.92322E−02	5.00000E−01	1.51000E−01	3.30013E−01	6.51650E−01	3.35398E+01
	SGD	4.27412E−01	1.19525E−01	4.01723E−01	1.67779E−01	2.63958E−01	6.51042E−01	**1.57049E+01**
6	VS-FNN	**2.48551E−01**	2.47933E−02	**2.34806E−01**	5.48683E−02	2.26495E−01	2.89343E−01	1.98839E+01
	PSO-FNN	2.51317E−01	2.64168E−02	2.39108E−01	5.62595E−02	**2.15496E−01**	**2.87191E−01**	2.01092E+01
	ABC-FNN	2.61312E−01	**1.83956E−02**	2.62269E−01	2.42259E−02	2.26098E−01	2.92864E−01	2.07117E+01
	GA-FNN	2.74379E−01	2.51352E−02	2.85257E−01	**2.19374E−02**	2.21772E−01	3.04955E−01	2.94405E+01
	SA-FNN	4.88994E−01	8.86623E−02	4.98047E−01	9.68635E−02	3.33624E−01	6.51020E−01	3.37511E+01
	SGD	4.16908E−01	8.71802E−02	4.43778E−01	1.50521E−01	2.89632E−01	5.66405E−01	**1.56219E+01**
7	VS-FNN	2.59204E−01	2.77429E−02	2.46604E−01	5.54473E−02	2.25775E−01	2.95378E−01	1.99857E+01
	PSO-FNN	**2.41422E−01**	2.52613E−02	**2.28573E−01**	**2.81092E−02**	2.14739E−01	**2.91755E−01**	1.98152E+01
	ABC-FNN	2.65226E−01	**2.03468E−02**	2.72106E−01	3.48303E−02	2.22840E−01	2.93661E−01	2.08298E+01
	GA-FNN	2.67934E−01	3.20405E−02	2.80677E−01	6.41614E−02	**2.08646E−01**	3.08057E−01	2.94103E+01
	SA-FNN	4.72386E−01	9.74111E−02	4.57965E−01	1.19642E−01	3.33772E−01	6.51042E−01	3.38716E+01
	SGD	4.40458E−01	8.05896E−02	4.55988E−01	1.25604E−01	2.95992E−01	6.50390E−01	**1.59233E+01**
8	VS-FNN	**2.57680E−01**	2.63870E−02	**2.48930E−01**	5.52433E−02	2.22502E−01	**2.94632E−01**	1.98573E+01
	PSO-FNN	2.63396E−01	2.78390E−02	2.83218E−01	5.60133E−02	**2.20073E−01**	2.99268E−01	1.98838E+01
	ABC-FNN	2.71140E−01	**1.66349E−02**	2.71101E−01	2.35445E−02	2.31178E−01	2.94895E−01	2.05719E+01
	GA-FNN	2.83008E−01	2.49745E−02	2.89209E−01	**1.59676E−02**	2.22698E−01	3.24944E−01	2.96856E+01
	SA-FNN	4.83734E−01	9.82966E−02	4.81130E−01	1.53766E−01	3.42394E−01	6.49371E−01	3.38740E+01
	SGD	4.47025E−01	9.75719E−02	4.75814E−01	1.67356E−01	3.04694E−01	6.19256E−01	**1.56181E+01**
9	VS-FNN	**2.58144E−01**	2.63434E−02	**2.57097E−01**	5.33926E−02	2.23081E−01	**2.92395E−01**	2.00348E+01
	PSO-FNN	2.64652E−01	2.86507E−02	2.80131E−01	5.33716E−02	2.18365E−01	3.07310E−01	2.00493E+01
	ABC-FNN	2.84932E−01	**1.41774E−02**	2.88014E−01	**1.66717E−02**	2.41513E−01	3.00649E−01	2.06882E+01
	GA-FNN	2.81710E−01	2.72651E−02	2.89409E−01	2.32139E−02	**2.09702E−01**	3.19141E−01	3.00711E+01
	SA-FNN	4.64946E−01	1.07771E−01	4.41292E−01	1.66403E−01	3.37755E−01	6.51042E−01	3.41440E+01
	SGD	4.47607E−01	8.57593E−02	4.75930E−01	1.52595E−01	3.11652E−01	6.51042E−01	**1.62089E+01**
10	VS-FNN	**2.62766E−01**	2.70677E−02	**2.60049E−01**	5.30943E−02	2.21530E−01	**2.96471E−01**	2.01852E+01
	PSO-FNN	2.71363E−01	2.36919E−02	2.83559E−01	3.44691E−02	2.23887E−01	3.08659E−01	2.01597E+01
	ABC-FNN	2.86274E−01	**1.52377E−02**	2.90256E−01	**1.52582E−02**	2.37171E−01	3.07649E−01	2.08977E+01
	GA-FNN	2.79553E−01	2.71543E−02	2.85158E−01	3.24312E−02	**2.19490E−01**	3.17895E−01	3.00387E+01
	SA-FNN	4.69302E−01	9.37816E−02	4.83080E−01	1.53604E−01	3.40927E−01	6.51042E−01	3.41315E+01
	SGD	4.58034E−01	9.85687E−02	4.68026E−01	1.54436E−01	3.02801E−01	6.49089E−01	**1.59013E+01**
11	VS-FNN	**2.67140E−01**	2.78631E−02	**2.83422E−01**	5.30660E−02	2.26003E−01	**3.10768E−01**	2.01077E+01
	PSO-FNN	2.83940E−01	2.00529E−02	2.85560E−01	**1.07398E−02**	2.27984E−01	3.14141E−01	2.01901E+01
	ABC-FNN	2.91032E−01	**1.16083E−02**	2.94743E−01	1.42581E−02	2.65354E−01	3.13070E−01	2.08730E+01
	GA-FNN	2.89250E−01	2.13633E−02	2.90545E−01	2.11975E−02	**2.14641E−01**	3.18298E−01	2.99172E+01
	SA-FNN	4.83129E−01	9.83437E−02	4.87947E−01	1.14739E−01	3.43910E−01	6.51690E−01	3.59008E+01
	SGD	4.31145E−01	8.86183E−02	4.35362E−01	1.42566E−01	2.89100E−01	6.33479E−01	**1.60619E+01**

Table 16 (continued)

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
13	VS-FNN	**2.85554E−01**	1.51011E−02	**2.87519E−01**	1.52785E−02	2.44562E−01	**3.08225E−01**	2.02720E+01
	PSO-FNN	2.88068E−01	2.08434E−02	2.89204E−01	1.68879E−02	**2.27380E−01**	3.19007E−01	2.04414E+01
	ABC-FNN	2.94442E−01	**1.24041E−02**	2.97712E−01	**1.24552E−02**	2.58601E−01	3.14652E−01	2.13059E+01
	GA-FNN	2.99240E−01	1.51098E−02	3.00569E−01	1.93312E−02	2.51644E−01	3.25551E−01	3.04375E+01
	SA-FNN	4.95651E−01	8.65188E−02	4.95551E−01	1.21835E−01	3.48958E−01	6.52988E−01	3.47484E+01
	SGD	4.38885E−01	9.59761E−02	4.48657E−01	1.59505E−01	3.13975E−01	6.51042E−01	**1.64357E+01**
15	VS-FNN	**2.88572E−01**	2.17788E−02	**2.90368E−01**	1.89749E−02	**2.26149E−01**	**3.16176E−01**	2.05970E+01
	PSO-FNN	2.90295E−01	2.06618E−02	2.91420E−01	2.28908E−02	2.36499E−01	3.21171E−01	2.06031E+01
	ABC-FNN	3.02924E−01	**9.24945E−03**	3.04074E−01	**1.30955E−02**	2.78199E−01	3.17365E−01	2.23360E+01
	GA-FNN	3.04385E−01	1.50171E−02	3.07721E−01	1.96697E−02	2.68207E−01	3.22954E−01	2.87666E+01
	SA-FNN	4.95524E−01	8.69944E−02	4.96794E−01	1.19693E−01	3.45105E−01	6.52342E−01	3.50361E+01
	SGD	4.82464E−01	1.12842E−01	4.86102E−01	1.82724E−01	3.10971E−01	6.51042E−01	**1.67870E+01**
20	VS-FNN	**2.92757E−01**	2.14124E−02	**2.94696E−01**	2.12155E−02	**2.40596E−01**	3.25740E−01	2.09752E+01
	PSO-FNN	3.03765E−01	1.56834E−02	3.08407E−01	2.10623E−02	2.62063E−01	3.26610E−01	2.10189E+01
	ABC-FNN	3.10563E−01	**1.05463E−02**	3.11174E−01	**7.09190E−03**	2.84208E−01	3.29203E−01	2.24659E+01
	GA-FNN	3.07650E−01	1.33853E−02	3.11082E−01	1.48770E−02	2.62784E−01	**3.24970E−01**	2.05256E+01
	SA-FNN	5.00747E−01	9.27795E−02	4.99674E−01	1.58647E−01	3.45831E−01	6.53637E−01	3.57134E+01
	SGD	4.57470E−01	8.88184E−02	4.59895E−01	1.15347E−01	3.23931E−01	6.51042E−01	**1.67066E+01**
30	VS-FNN	**3.05643E−01**	1.17529E−02	**3.06655E−01**	1.50655E−02	**2.75004E−01**	**3.22677E−01**	2.18627E+01
	PSO-FNN	3.10220E−01	8.69532E−03	3.11862E−01	1.05470E−02	2.85220E−01	3.27671E−01	2.17731E+01
	ABC-FNN	3.19344E−01	**6.70109E−03**	3.18926E−01	**8.16108E−03**	3.06368E−01	3.33369E−01	2.30122E+01
	GA-FNN	3.14991E−01	8.86047E−03	3.14732E−01	1.32024E−02	2.96195E−01	3.28493E−01	1.81219E+01
	SA-FNN	4.94327E−01	9.30034E−02	5.00050E−01	1.64306E−01	3.47709E−01	6.51042E−01	3.68434E+01
	SGD	4.73566E−01	7.73720E−02	4.94422E−01	9.15384E−02	3.39183E−01	6.52344E−01	**1.75909E+01**

The best results are indicated in bold

Table 17 The comparison of the ranks of the algorithms for the results of PID benchmark problem with respect to average MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	10	1	1	0	0	0
PSO-FNN	2	10	0	0	0	0
ABC-FNN	0	1	6	5	0	0
GA-FNN	0	0	5	7	0	0
SA-FNN	0	0	0	0	0	12
SGD	0	0	0	0	12	0

Table 18 The comparison of the ranks of the algorithms for the results of PID benchmark problem with respect to median MSE

Metho	Orders					
	1	2	3	4	5	6
VS-FNN	10	2	0	0	0	0
PSO-FNN	2	9	1	0	0	0
ABC-FNN	0	1	7	4	0	0
GA-FNN	0	0	4	8	0	0
SA-FNN	0	0	0	0	1	11
SGD	0	0	0	0	11	1

Table 19 The statistical results of MSE over 30 independent runs for the Thyroid benchmark problem

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
4	VS-FNN	4.71404E−02	1.10702E−03	4.74341E−02	7.36994E−04	4.18557E−02	**4.85655E−02**	**1.93774E+02**
	PSO-FNN	4.77126E−02	9.47588E−04	4.76310E−02	1.28537E−03	4.51266E−02	4.92478E−02	3.34322E+02
	ABC-FNN	4.89608E−02	**5.17040E−04**	4.91339E−02	**5.72782E−04**	4.73360E−02	4.94909E−02	2.44621E+02
	GA-FNN	**4.67458E−02**	1.22168E−03	**4.66180E−02**	1.60639E−03	4.34014E−02	4.90263E−02	2.87138E+02
	SA-FNN	4.81811E−01	2.29771E−01	4.52579E−01	3.16452E−01	5.23943E−02	9.49756E−01	4.06411E+02
	SGD	1.76914E−01	1.62854E−01	5.84265E−02	2.85704E−01	**3.67246E−02**	6.31090E−01	2.61521E+02
5	VS-FNN	4.76291E−02	8.90444E−04	**4.76599E−02**	1.13018E−03	4.51049E−02	**4.90530E−02**	**1.93117E+02**
	PSO-FNN	4.78289E−02	1.32378E−03	4.83115E−02	1.73924E−03	**4.38804E−02**	4.93167E−02	3.38653E+02
	ABC-FNN	4.90587E−02	**4.54149E−04**	4.92195E−02	**5.05794E−04**	4.77550E−02	4.95849E−02	2.51098E+02
	GA-FNN	**4.75950E−02**	9.31677E−04	4.76661E−02	1.37745E−03	4.51005E−02	4.92234E−02	2.96641E+02
	SA-FNN	5.20093E−01	2.00958E−01	5.13989E−01	2.54506E−01	7.09976E−02	9.50507E−01	3.69887E+02
	SGD	1.91171E−01	1.49823E−01	9.16810E−02	2.89643E−01	4.68965E−02	3.67630E−01	2.63597E+02
6	VS-FNN	4.76384E−02	1.12467E−03	**4.76935E−02**	1.45157E−03	4.43616E−02	**4.93250E−02**	**1.94535E+02**
	PSO-FNN	4.82995E−02	9.91689E−04	4.86176E−02	1.21934E−03	4.58880E−02	4.93749E−02	3.30086E+02
	ABC-FNN	4.92326E−02	**4.15323E−04**	4.93158E−02	**2.55089E−04**	4.77711E−02	5.00485E−02	2.60582E+02
	GA-FNN	**4.71775E−02**	2.04535E−03	4.78015E−02	2.27360E−03	**3.92140E−02**	4.93332E−02	2.92781E+02
	SA-FNN	4.53670E−01	1.76128E−01	4.59287E−01	2.41956E−01	9.29216E−02	8.41741E−01	4.35383E+02
	SGD	2.18713E−01	2.16409E−01	6.95721E−02	3.01612E−01	4.35766E−02	9.50507E−01	2.68846E+02
7	VS-FNN	4.80945E−02	1.04958E−03	4.85013E−02	1.20755E−03	4.52845E−02	4.93999E−02	**1.96869E+02**
	PSO-FNN	4.80023E−02	1.31827E−03	4.82809E−02	1.54781E−03	4.42593E−02	**4.93098E−02**	3.19479E+02
	ABC-FNN	4.93916E−02	**3.34440E−04**	4.94281E−02	**2.37540E−04**	4.88155E−02	5.08517E−02	2.59125E+02
	GA-FNN	**4.75237E−02**	2.16655E−03	**4.81771E−02**	2.23142E−03	3.97355E−02	4.94393E−02	2.96185E+02
	SA-FNN	5.02770E−01	2.34775E−01	4.58702E−01	2.76999E−01	1.00285E−01	9.41573E−01	5.92104E+02
	SGD	1.09961E−01	1.21903E−01	4.99597E−02	1.06845E−02	**3.45873E−02**	3.68320E−01	2.52892E+02
8	VS-FNN	4.78295E−02	1.24347E−03	**4.80488E−02**	1.79188E−03	4.52468E−02	**4.93785E−02**	**1.97299E+02**
	PSO-FNN	4.81143E−02	1.78147E−03	4.88010E−02	1.33808E−03	4.27545E−02	4.94309E−02	3.58066E+02
	ABC-FNN	4.95676E−02	**9.44762E−04**	4.94800E−02	**1.70750E−04**	4.84897E−02	5.34740E−02	2.55988E+02
	GA-FNN	**4.78164E−02**	2.01580E−03	4.85553E−02	1.28377E−03	4.16353E−02	4.94586E−02	2.98992E+02
	SA-FNN	4.69498E−01	2.19136E−01	4.09630E−01	3.25716E−01	7.04170E−02	9.16504E−01	5.60086E+02
	SGD	1.40219E−01	1.41910E−01	5.00667E−02	2.83959E−01	**3.82029E−02**	3.68531E−01	2.55821E+02
9	VS-FNN	**4.73915E−02**	1.61031E−03	**4.77257E−02**	2.04569E−03	4.31885E−02	**4.93999E−02**	**1.98979E+02**
	PSO-FNN	4.83208E−02	**1.11835E−03**	4.86943E−02	1.50417E−03	4.40093E−02	4.93999E−02	4.45268E+02
	ABC-FNN	5.05192E−02	6.62897E−03	4.94552E−02	**2.87249E−04**	4.74931E−02	8.55291E−02	2.44297E+02
	GA-FNN	4.82135E−02	1.40883E−03	4.87943E−02	1.53284E−03	4.34328E−02	4.94272E−02	3.00164E+02
	SA-FNN	5.55639E−01	2.19006E−01	6.23382E−01	2.62835E−01	1.22348E−01	9.49786E−01	6.75877E+02
	SGD	1.16077E−01	1.46291E−01	4.95681E−02	2.08733E−03	**3.76420E−02**	6.31725E−01	2.51765E+02
10	VS-FNN	**4.75330E−02**	1.90277E−03	4.84151E−02	2.58747E−03	4.33283E−02	4.93485E−02	**2.08354E+02**
	PSO-FNN	4.85932E−02	**7.67109E−04**	4.89944E−02	1.12561E−03	4.66822E−02	**4.92984E−02**	4.50877E+02
	ABC-FNN	4.97151E−02	8.26286E−04	4.94926E−02	**1.31851E−04**	4.87056E−02	5.35613E−02	2.47644E+02
	GA-FNN	4.77134E−02	1.85475E−03	**4.82488E−02**	1.88792E−03	**4.23833E−02**	4.93982E−02	2.91129E+02
	SA-FNN	4.94603E−01	1.95751E−01	5.36151E−01	2.71943E−01	4.98497E−02	8.43492E−01	7.69239E+02
	SGD	1.13949E−01	1.44680E−01	5.02931E−02	2.62088E−03	4.80283E−02	6.63237E−01	3.72712E+02
11	VS-FNN	**4.72455E−02**	1.77323E−03	**4.75828E−02**	1.90939E−03	4.17358E−02	**4.93276E−02**	**2.15727E+02**
	PSO-FNN	4.84068E−02	**1.08001E−03**	4.89153E−02	1.35527E−03	4.50237E−02	4.94343E−02	3.74746E+02
	ABC-FNN	4.99142E−02	1.29989E−03	4.94926E−02	**3.47480E−04**	4.88984E−02	5.59374E−02	2.26012E+02
	GA-FNN	4.77249E−02	2.00511E−03	4.85709E−02	1.79715E−03	4.29719E−02	4.94439E−02	2.92933E+02
	SA-FNN	4.89273E−01	2.21237E−01	4.94386E−01	2.69266E−01	1.09856E−01	9.37824E−01	5.19028E+02
	SGD	1.09128E−01	1.19854E−01	5.03312E−02	6.25184E−03	**4.00072E−02**	3.66839E−01	3.67861E+02

Table 19 (continued)

Hidden nodes	Algorithm	Average MSE	Std dev MSE	Median MSE	IQR	Best MSE	Worst MSE	Average time
13	VS-FNN	**4.76812E−02**	1.76798E−03	4.83250E−02	2.23946E−03	4.41168E−02	4.93955E−02	3.56602E+02
	PSO-FNN	4.84716E−02	**1.15150E−03**	4.87102E−02	**1.28636E−03**	4.37719E−02	4.93976E−02	2.93600E+02
	ABC-FNN	5.32170E−02	7.17577E−03	4.99839E−02	3.64663E−03	4.94000E−02	8.20546E−02	**2.30340E+02**
	GA-FNN	4.78997E−02	1.45501E−03	**4.82923E−02**	1.81576E−03	4.43856E−02	**4.93548E−02**	2.99020E+02
	SA-FNN	4.62943E−01	2.10099E−01	4.83360E−01	3.35284E−01	7.09168E−02	8.62527E−01	3.85633E+02
	SGD	8.97659E−02	1.02266E−01	5.04659E−02	4.96322E−03	**4.07766E−02**	3.51282E−01	3.72727E+02
15	VS-FNN	4.80744E−02	1.37961E−03	4.83146E−02	2.27889E−03	4.44951E−02	4.94462E−02	3.68395E+02
	PSO-FNN	4.80056E−02	**1.07700E−03**	**4.83116E−02**	**1.38980E−03**	4.47919E−02	**4.93535E−02**	4.52181E+02
	ABC-FNN	5.14860E−02	4.20503E−03	4.96306E−02	1.89749E−03	4.86755E−02	6.88157E−02	**2.59645E+02**
	GA-FNN	**4.78386E−02**	1.87877E−03	4.84823E−02	2.33526E−03	4.18324E−02	4.94462E−02	2.95349E+02
	SA-FNN	5.04542E−01	1.92979E−01	5.44986E−01	2.62541E−01	4.96790E−02	7.62596E−01	7.59223E+02
	SGD	8.83957E−02	1.29509E−01	5.02519E−02	3.34278E−03	**3.70976E−02**	6.51597E−01	3.92733E+02
20	VS-FNN	**4.76261E−02**	1.76170E−03	**4.84989E−02**	2.94825E−03	4.42278E−02	4.94462E−02	3.62477E+02
	PSO-FNN	4.84445E−02	**1.10757E−03**	4.88571E−02	**1.12535E−03**	4.50259E−02	4.94462E−02	4.65061E+02
	ABC-FNN	5.66235E−02	1.74468E−02	5.03604E−02	8.03958E−03	4.93635E−02	1.43119E−01	3.69830E+02
	GA-FNN	4.78460E−02	2.05308E−03	4.89747E−02	2.57821E−03	4.28867E−02	**4.94129E−02**	**3.07315E+02**
	SA-FNN	5.06372E−01	2.06966E−01	5.56773E−01	3.50292E−01	7.32396E−02	8.27919E−01	7.02485E+02
	SGD	1.22590E−01	1.28622E−01	5.06252E−02	9.03741E−02	**3.87301E−02**	3.73929E−01	4.09527E+02
30	VS-FNN	4.85657E−02	**1.13558E−03**	4.91683E−02	**1.20371E−03**	4.43433E−02	**4.94048E−02**	3.78630E+02
	PSO-FNN	4.84149E−02	1.38943E−03	**4.91619E−02**	1.42410E−03	4.40736E−02	4.94462E−02	4.42612E+02
	ABC-FNN	6.21097E−02	1.95818E−02	5.30433E−02	1.33967E−02	4.94517E−02	1.31081E−01	3.97514E+02
	GA-FNN	**4.82085E−02**	1.73692E−03	4.91812E−02	2.24317E−03	**4.37047E−02**	4.94462E−02	**3.13795E+02**
	SA-FNN	4.44210E−01	1.97044E−01	4.07539E−01	2.49987E−01	6.55532E−02	8.72497E−01	4.89074E+02
	SGD	6.14623E−02	5.35528E−02	5.04080E−02	1.71450E−03	4.86106E−02	3.44224E−01	4.31097E+02

The best results are indicated in bold

Table 20 The comparison of the ranks of the algorithms for the results of Thyroid problem with respect to average MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	5	4	3	0	0	0
PSO-FNN	0	3	9	0	0	0
ABC-FNN	0	0	0	11	1	0
GA-FNN	7	5	0	0	0	0
SA-FNN	0	0	0	0	0	12
SGD	0	0	0	1	11	0

Table 21 The comparison of the ranks of the algorithms for the results of Thyroid problem with respect to median MSE

Method	Orders					
	1	2	3	4	5	6
VS-FNN	6	5	1	0	0	0
PSO-FNN	2	3	7	0	0	0
ABC-FNN	0	0	0	11	1	0
GA-FNN	4	4	4	0	0	0
SA-FNN	0	0	0	0	0	12
SGD	0	0	0	1	11	0

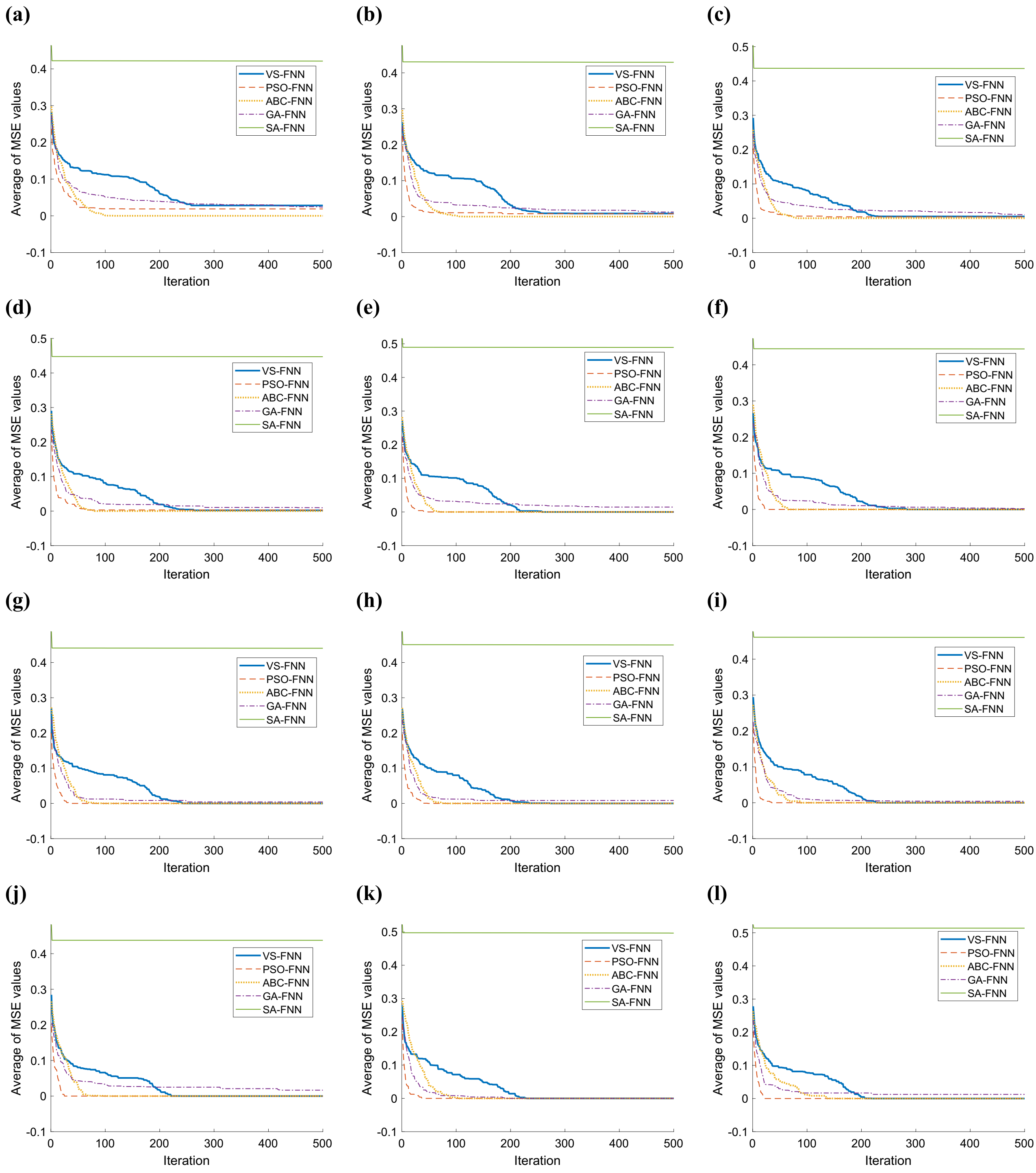

Fig. 6 Convergence curves of VS-FNN, PSO-FNN, ABC-FNN, GA-FNN, and SA-FNN based on averages of MSE values over 30 independent runs in a 3-bit XOR problem. **a**–**l** are the convergence curves for FNNs with H = 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 20, and 30, respectively

and ABC in terms of duration. Figure 8 shows the convergence curves of all algorithms based on the average of MSE values. These figures confirm that VS-FNN has a remarkable performance with a competitive convergence rate for all FNNs.

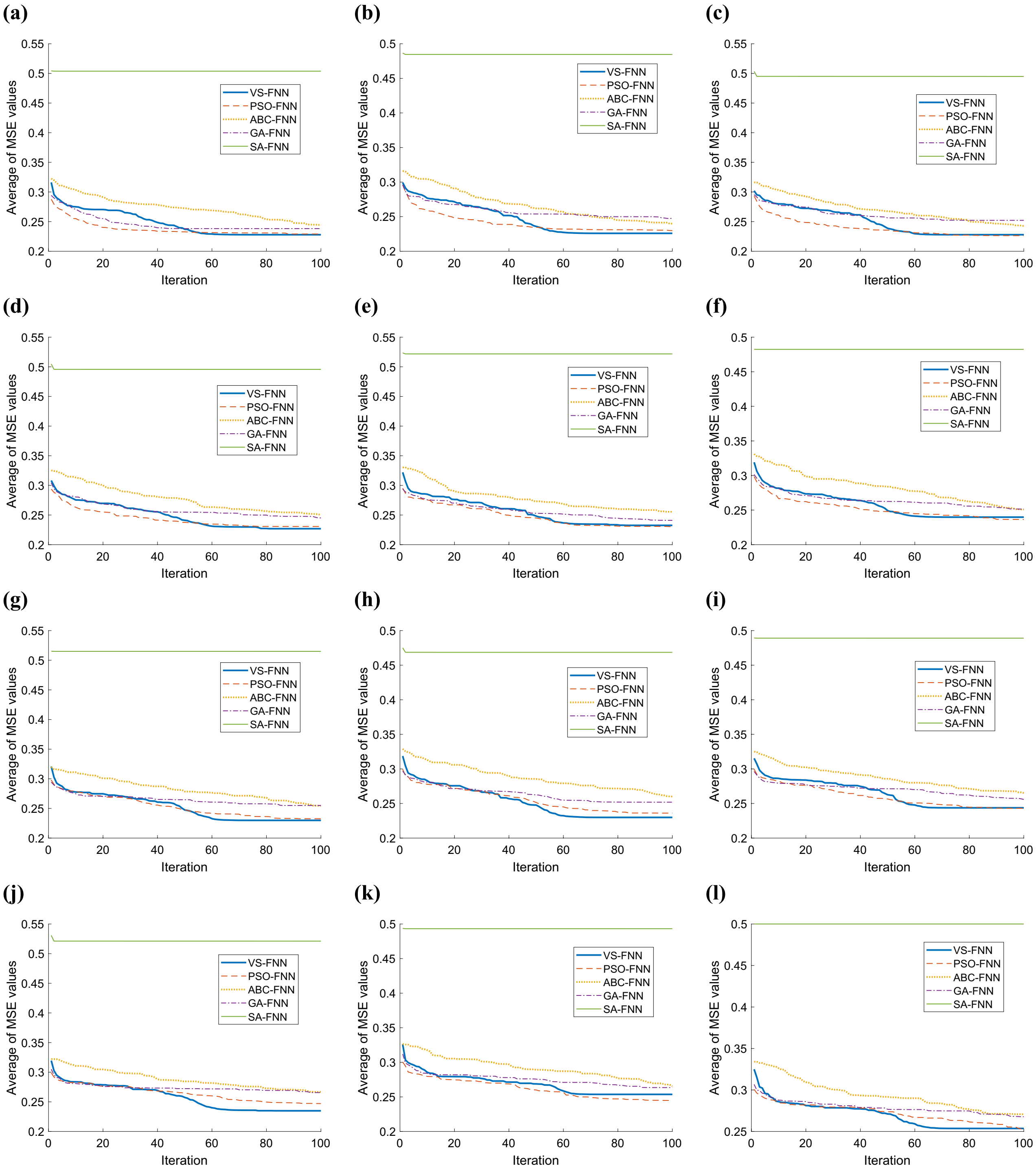

Fig. 7 Convergence curves of VS-FNN, PSO-FNN, ABC-FNN, GA-FNN, and SA-FNN based on averages of MSE values over 30 independent runs in the Iris benchmark problem. **a–l** are the convergence curves for FNNs with H = 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 20, and 30, respectively

6.2.4 The WBC problem

The Wisconsin-Breast-Cancer (Original) dataset is another well-studied benchmarking problem that contains the samples of the most common malignancy among women to identify into benign or malignant classes [20]. FNNs with structures 10–H–2 are trained by the algorithms. The comparative results of FNNs to solve this problem are presented in Table 13. Apparently, for all hidden nodes, VS-FNN has the best values for the median and IQR. With

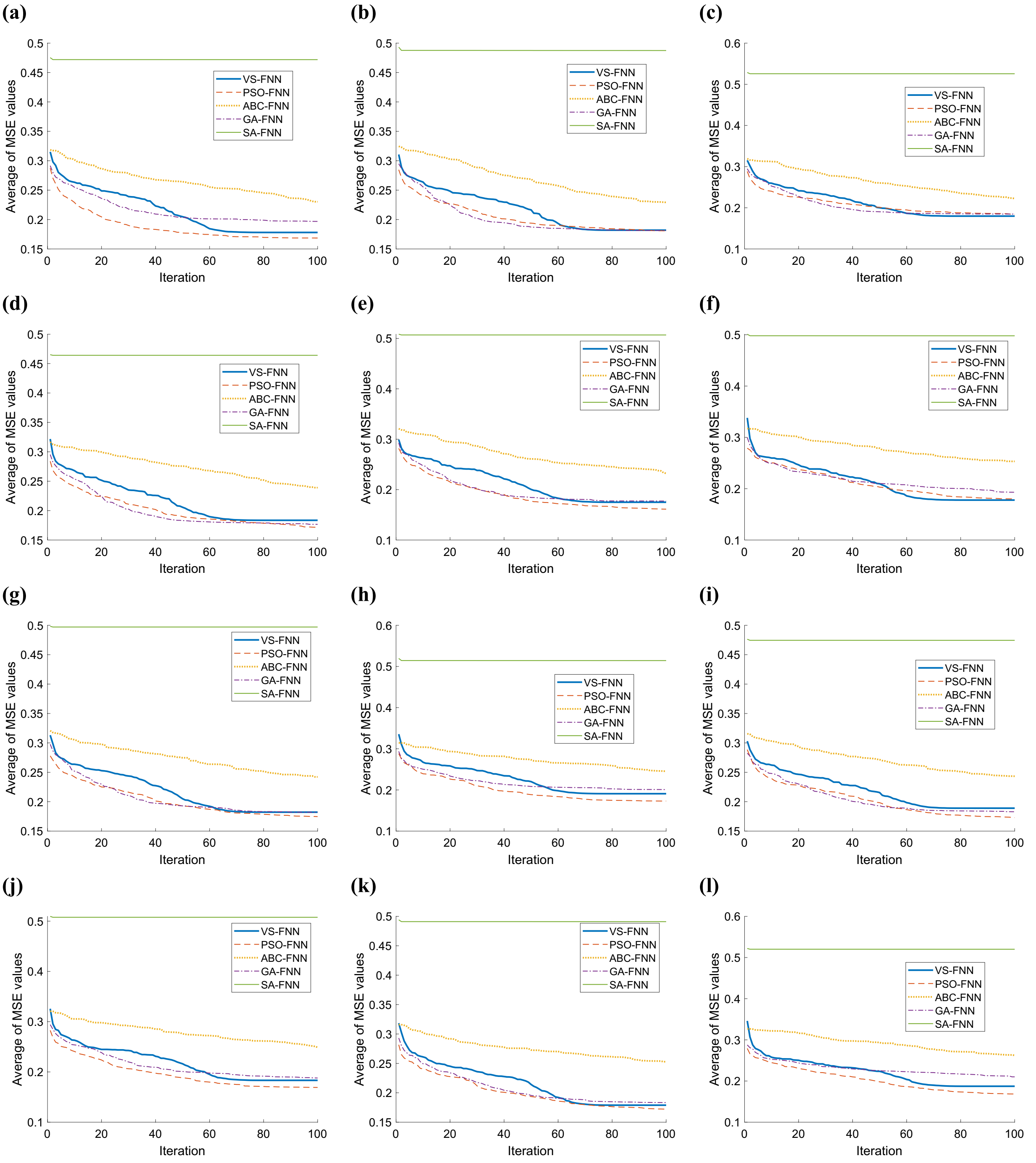

Fig. 8 Convergence curves of VS-FNN, PSO-FNN, ABC-FNN, GA-FNN, and SA-FNN based on averages of MSE values over 30 independent runs in the Wine benchmark problem. **a**–**l** are the convergence curves for FNNs with H = 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 20, and 30, respectively

regard to the values of average and standard deviation, VS-FNN and PSO-FNN have very close and the best values together. This is seen from Tables 14 and 15. The number of the first ranks for average MSE values is equal to each other for VS-FNN and PSO-FNN while the number of first ranks for median MSE values is 12 for VS-FNN. These

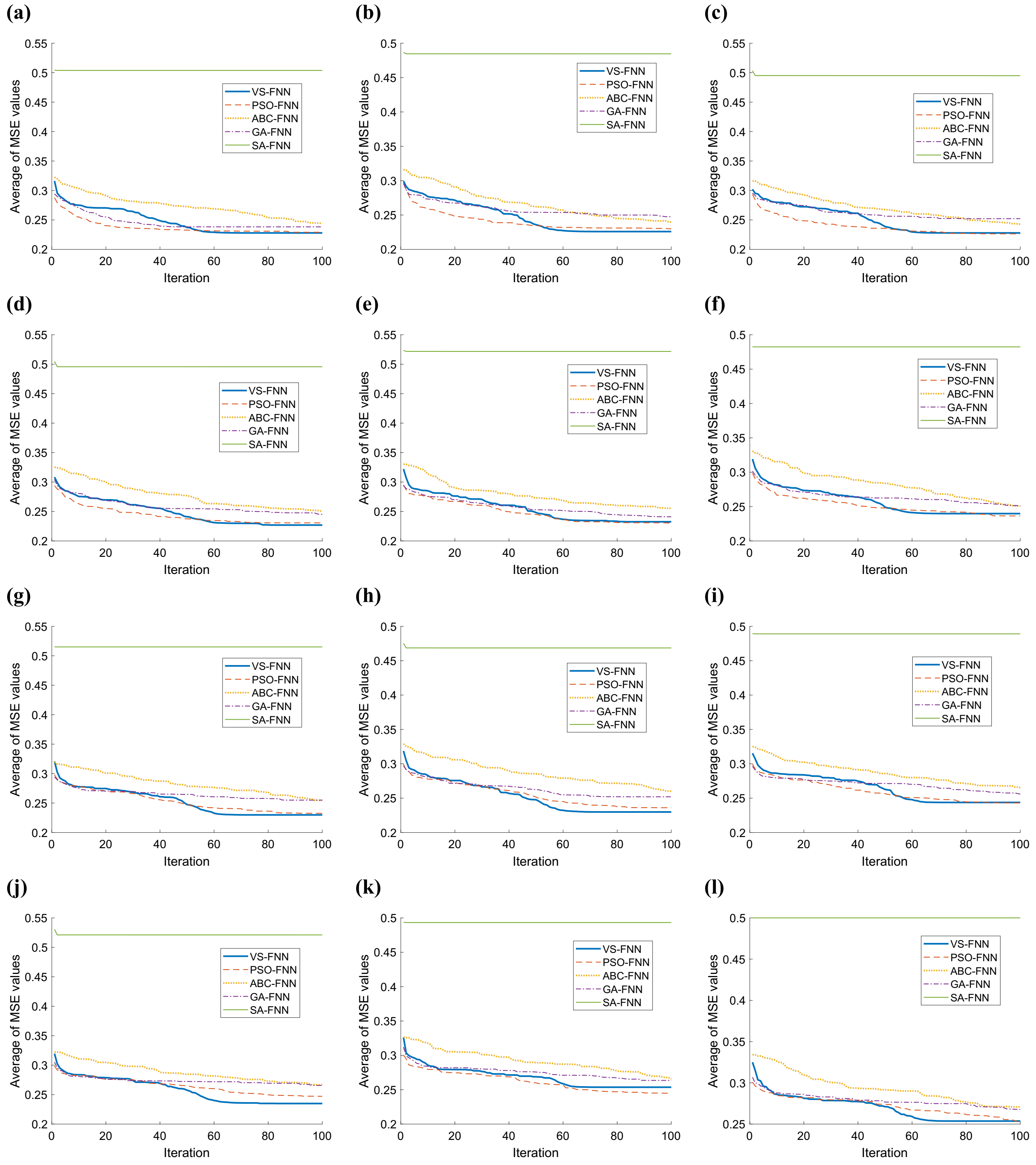

Fig. 9 Convergence curves of VS-FNN, PSO-FNN, ABC-FNN, GA-FNN, and SA-FNN based on averages of MSE values over 30 independent runs in the WBC benchmark problem. **a**–**l** are the convergence curves for FNNs with H = 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 20, and 30, respectively

results prove that the proposed VS based learning method can reach the most accurate results and avoid local minima for this problem. Figure 9 shows the convergence curves of all algorithms for the WBC problem. These figures confirm that VS-FNN has the best convergence rate for almost all values of hidden nodes.

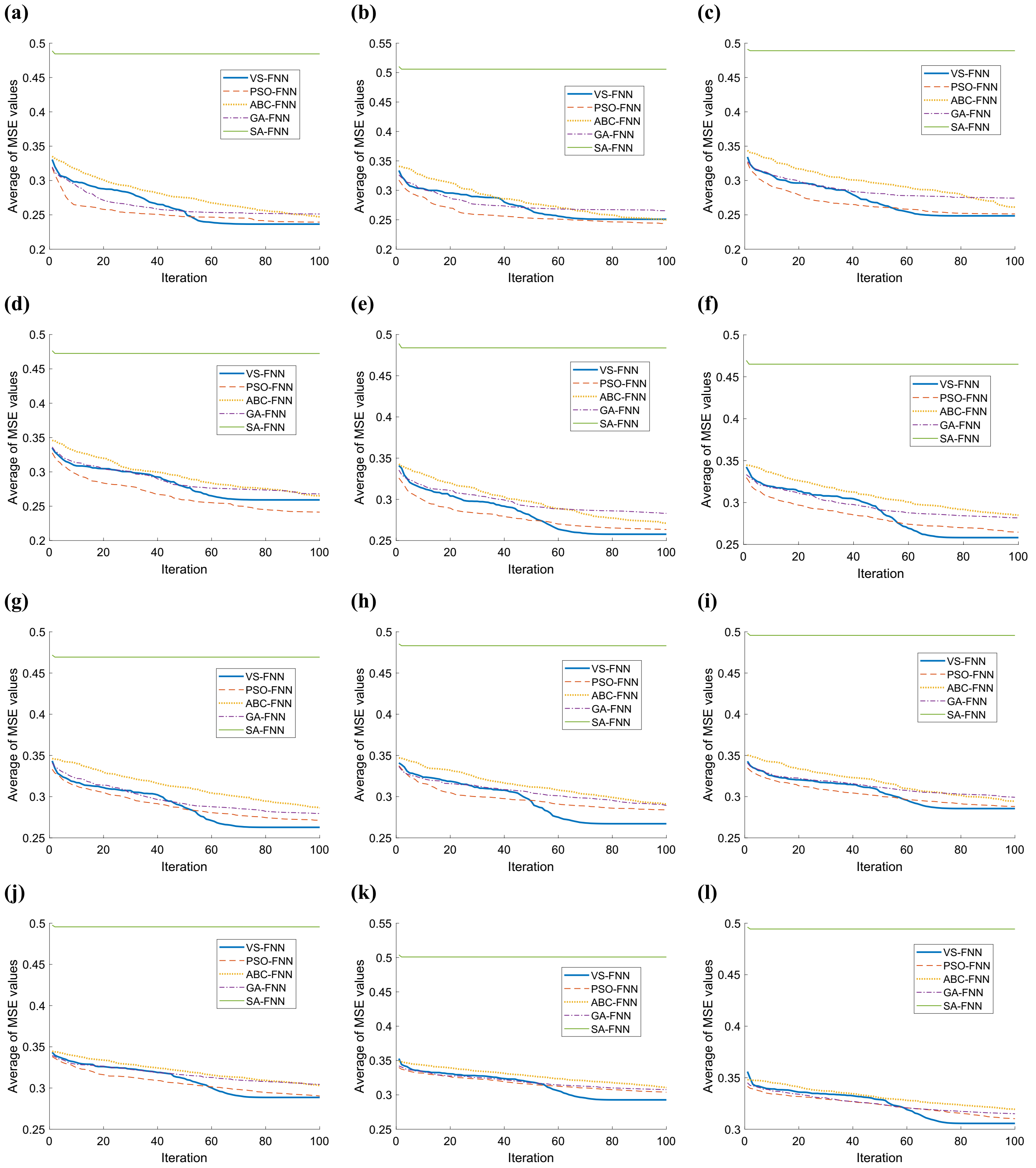

Fig. 10 Convergence curves of VS-FNN, PSO-FNN, ABC-FNN, GA-FNN, and SA-FNN based on averages of MSE values over 30 independent runs in the PID benchmark problem. **a**–**l** are the convergence curves for FNNs with H = 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 20, and 30, respectively

6.2.5 The PID problem

The Pima-Indians-Diabetes dataset consists of the diagnostic measurements of some patients who are females at least 21 years old of Pima Indian heritage. With this dataset, a classification is made to predict whether a patient has diabetes or not. The PID dataset has been frequently used as a benchmark in the field of machine learning. In this study,

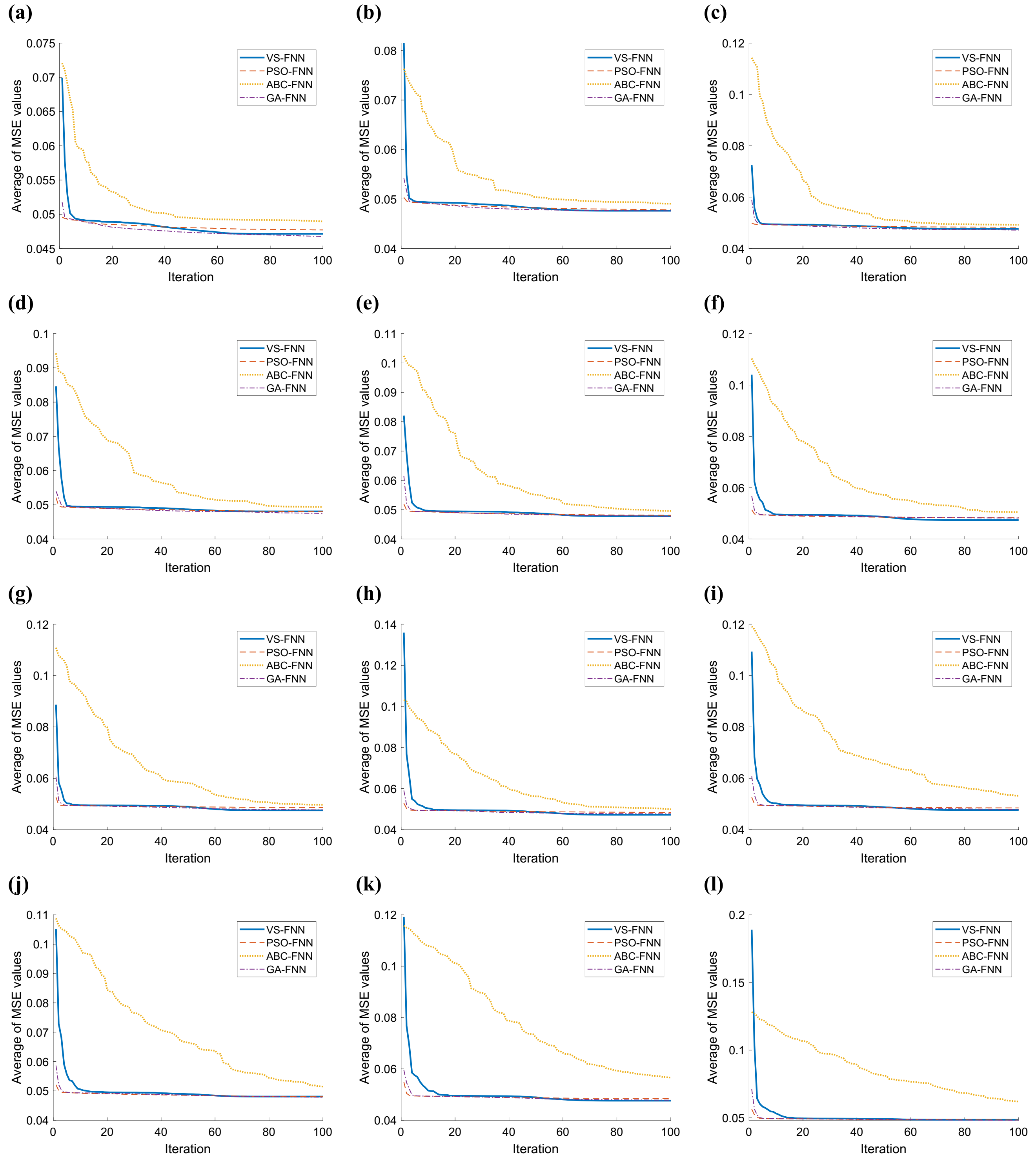

Fig. 11 Convergence curves of VS-FNN, PSO-FNN, ABC-FNN, and GA-FNN based on averages of MSE values over 30 independent runs in the Thyroid benchmark problem. **a**–**l** are the convergence curves for FNNs with H = 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 20, and 30, respectively

FNNs with structures 8–H–2 are trained by the algorithms. Table 16 shows the statistical results of MSE values over 30 independent runs for the PID problem. Tables 17 and 18 provide the rank-based comparisons for the PID problem with respect to average and median values of MSE, respectively. From these tables, it can be seen that VS-FNN performs better than the other algorithms. Both average and median MSE values are superior in 10 of 12 FNNs. In almost all structures, VS-FNN achieves more accurate values than others. Figure 10 shows the convergence curves of all algorithms based on the average of MSE values for the PID problem. These figures reflect the balance ability of VS-FNN between exploration and exploitation. It has a weak locality until the 50th iteration. After the half of the maximum number of iterations, the radius decreases significantly so that the algorithm has a strong locality.

6.2.6 The thyroid-disease problem

The Thyroid-Disease dataset is one of the most commonly used datasets for classification systems in machine learning literature. It contains the samples to classify that a given patient is normal (i) or suffers from hyperthyroidism (ii) or hypothyroidism (iii). For this purpose, FNNs with structures 21–H–3 are trained by the algorithms. The statistical results for the thyroid-disease benchmark problem are presented in Table 19. It can be seen the VS-FNN performs a competitive learning process with GA-FNN and PSO-FNN methods. The VS-FNN has better results in many of 12 different FNN structures. The rank-based comparisons for the thyroid-disease problem are provided in Tables 20 and 21 according to the average and median of MSE, respectively. As can be inferred from these tables, the proposed method achieves one of the best rankings among the other state-of-the-art algorithms. Figure 11 shows the convergence curves of all algorithms, except SA, based on the average of MSE values. The results of the SA are not included in the figures so that the curve of other algorithms can be seen in more detail. From these figures, it is seen that all algorithms except ABC present a close convergence rate to each other.

7 Conclusions

Neural Networks has always been one of the most important topics in machine learning, and FNN is the most popular type of neural network in many applications. The success of FNNs used for classification and prediction on many real-world problems depends largely on the correct training of the network and the performance of the training method. Derivative classical learning algorithms are insufficient in the training of FNNs due to factors such as non-linearity of problem types and very large dimensions. In the literature, many metaheuristic optimization algorithms have been proposed to determine the optimal values of weights for the solution of such problems.

The main contribution of this paper is that it is the first attempt to investigate the performance of the Vortex Search optimization algorithm in the process of FNN training. The VS algorithm deserves this thanks to its noteworthy performance and features such as not needing any user-tuning parameters, low computational complexity, fast convergence, and including a simple but effective search method based on Gaussian distribution.

In this study, the VS algorithm is adapted into the training process of FNNs as a new training method called VS-FNN. The performance of the proposed method was compared with six other FNNs trained with SGD, ABC, PSO, SA, and GA. Under the same running conditions, all algorithms are applied to six benchmark problems: three-bits parity (3-bit XOR) problem, Iris classification, Wine-Recognition, Wisconsin-Breast-Cancer, Pima-Indians-Diabetes, and Thyroid-Disease. To compare the effectiveness of the proposed algorithm in more detail, all datasets were run for neural network structures containing twelve different numbers of hidden nodes. The experimental results obtained after 30 independent runs prove that the proposed training method shows better or competitive performance in terms of accuracy, convergence rate, and computation time for all benchmark problems. Therefore, it can be concluded that the VS-FNN method a suitable algorithm for the training of FNNs.

References

1. Abiodun OI, Jantan A, Omolara AE, Dada KV, Mohamed NA, Arshad H (2018) State-of-the-art in artificial neural network applications: a survey Heliyon 4. https://doi.org/10.1016/j.heliyon.2018.e00938
2. Annema J (1995) Feed-forward neural networks. In: The Springer International Series in Engineering and Computer Science, vol 314. Springer, New York. https://doi.org/10.1007/978-1-4615-2337-6
3. Arora S, Singh S (2015) Butterfly Algorithm with Levy Flights for Global Optimization. In: 2015 International Conference on Signal Processing, Computing and Control (ISPCC), Waknaghat, India, 24–26 Sept. 2015. IEEE, pp 220–224
4. Askarzadeh A, Rezazadeh A (2013) Artificial neural network training using a new efficient optimization algorithm Appl. Soft Comput 13:1206–1213. https://doi.org/10.1016/j.asoc.2012.10.023
5. Battiti R (1992) First- and second-order methods for learning: between steepest descent and Newton's method. Neural Comput 4:141–166. https://doi.org/10.1162/neco.1992.4.2.141
6. Biron PV (1997) Backpropagation: theory, architectures, and applications - Chauvin Y, Rumelhart, DE. J Am Soc Inform Sci 48:88–89
7. Brajevic I, Tuba M Training feed-forward neural networks using firefly algorithm. In: Proceedings of the 12th International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'13), 2013. pp 156–161
8. Cao WP, Wang XZ, Ming Z, Gao JZ (2018) A review on neural networks with random weights. Neurocomputing 275:278–287. https://doi.org/10.1016/j.neucom.2017.08.040
9. Dogan B, Olmez T (2015) A new metaheuristic for numerical function optimization: vortex search algorithm information sciences 293:125–145 https://doi.org/10.1016/j.ins.2014.08.053
10. Frank A, Asuncion A (2010) Uci machine learning repository Irvine. University of California, School of Information and Computer Science, CA, p 21

11. Gudise VG, Venayagamoorthy GK (2003) Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS03), Indianapolis, IN, USA, USA, 26–26 April 2003. IEEE, pp 110–117. https://doi.org/10.1109/Sis.2003.1202255
12. Hagan MT, Menhaj MB (1994) Training feedforward networks with the marquardt algorithm. IEEE T Neural Networ 5:989–993. https://doi.org/10.1109/72.329697
13. Han F, Jiang J, Ling QH, Su BY (2019) A survey on metaheuristic optimization for random single-hidden layer feedforward neural network. Neurocomputing 335:261–273. https://doi.org/10.1016/j.neucom.2018.07.080
14. Holland JH (1984) Genetic Algorithms and Adaptation. In: Selfridge OG, Rissland EL, Arbib MA (eds) Adaptive Control of Ill-Defined Systems. Springer, Boston, pp 317–333. https://doi.org/10.1007/978-1-4684-8941-5_21
15. Ince T, Kiranyaz S, Pulkkinen J, Gabbouj M (2010) Evaluation of global and local training techniques over feed-forward neural network architecture spaces for computer-aided medical diagnosis. Expert Syst Appl 37:8450–8461. https://doi.org/10.1016/j.eswa.2010.05.033
16. Karaboga D, Akay B Artificial bee colony (ABC) algorithm on training artificial neural networks. In: 2007 IEEE 15th Signal Processing and Communications Applications, Eskisehir, Turkey, 2007. IEEE, pp 818–821. doi:https://doi.org/10.1109/SIU.2007.4298679
17. Kaur G, Arora S (2018) Chaotic whale optimization algorithm. J Comput Design Eng 5:275–284. https://doi.org/10.1016/j.jcde.2017.12.006
18. Kennedy J, Eberhart R (1995) Particle Swarm Optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, Australia, 27 Nov.-1 Dec. 1995. IEEE, pp 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
19. Kirkpatrick S, Gelatt CD, Vecchi MP (1987) Optimization by Simulated Annealing. In: Fischler MA, Firschein O (eds) Readings in Computer Vision. Morgan Kaufmann, San Francisco, pp 606–615. https://doi.org/10.1016/B978-0-08-051581-6.50059-3
20. Mangasarian OL, Wolberg WH (1990) Cancer diagnosis via linear programming. University of Wisconsin-Madison Department of Computer Sciences
21. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
22. Mirjalili S, Mohd Hashim SZ, Moradian Sardroudi H (2012) Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. Appl Math Comput 218:11125–11137. https://doi.org/10.1016/j.amc.2012.04.069
23. Mirjalili SZ, Saremi S, Mirjalili SM (2015) Designing evolutionary feedforward neural networks using social spider optimization algorithm. Neural Comput Appl 26:1919–1928. https://doi.org/10.1007/s00521-015-1847-6
24. Mohandes SR, Zhang X, Mahdiyar A (2019) A comprehensive review on the application of artificial neural networks in building energy analysis. Neurocomputing 340:55–75. https://doi.org/10.1016/j.neucom.2019.02.040
25. Montana DJ, Davis L (1989) Training feedforward neural networks using genetic algorithms. In: IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence, Detroit, Michigan, August 20 - 25, 1989. Morgan Kaufmann Publishers Inc., pp 762–767
26. Ozturk C, Karaboga D (2011) Hybrid Artificial Bee Colony Algorithm for Neural Network Training. In: 2011 IEEE Congress on Evolutionary Computation (CEC), New Orleans, LA, USA. IEEE, pp 84–88. https://doi.org/10.1109/CEC.2011.5949602
27. Piotrowski AP (2014) Differential evolution algorithms applied to neural network training suffer from stagnation. Appl Soft Comput 21:382–406. https://doi.org/10.1016/j.asoc.2014.03.039
28. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: A Gravitational Search Algorithm. Inf Sci 179:2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
29. Robbins H, Monro S (1951) A stochastic approximation method. Ann Math Stat 22:400–407
30. Simon D (2008) Biogeography-based optimization. IEEE T Evolut Comput 12:702–713. https://doi.org/10.1109/Tevc.2008.919004
31. Socha K, Blum C (2007) An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. Neural Comput Appl 16:235–247. https://doi.org/10.1007/s00521-007-0084-z
32. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359. https://doi.org/10.1023/A:1008202821328
33. Swain RR, Khilar PM, Dash T (2020) Multifault diagnosis in WSN using a hybrid metaheuristic trained neural network. Digit Commun Netw 6:86–100. https://doi.org/10.1016/j.dcan.2018.02.001
34. Tang R, Fong S, Deb S, Vasilakos AV, Millham RC (2018) Dynamic group optimisation algorithm for training feed-forward neural networks. Neurocomputing 314:1–19. https://doi.org/10.1016/j.neucom.2018.03.043
35. Treadgold NK, Gedeon TD (1998) Simulated annealing and weight decay in adaptive learning: The SARPROP algorithm. IEEE T Neural Networ 9:662–668. https://doi.org/10.1109/72.701179
36. van der Smagt PP (1994) Minimization methods for training feedforward neural networks. Neural Netw 7:1–11
37. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization IEEE transactions on evolutionary computation 1:67–82 https://doi.org/10.1109/4235.585893
38. Xu F, Pun C-M, Li H, Zhang Y, Song Y, Gao H (2019) Training feed-forward artificial neural networks with a modified artificial bee colony algorithm. Neurocomputing. https://doi.org/10.1016/j.neucom.2019.04.086
39. Yaghini M, Khoshraftar MM, Fallahi M (2013) A hybrid algorithm for artificial neural network training. Eng Appl Artif Intel 26:293–301. https://doi.org/10.1016/j.engappai.2012.01.023
40. Yang XS (2009) Firefly algorithms for multimodal optimization stochastic algorithms: foundations and applications SAGA 2009 lecture notes in computer science 5792:169–178. https://doi.org/10.1007/978-3-642-04944-6_14
41. Zhang J-R, Zhang J, Lok T-M, Lyu MR (2007) A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Appl Math Comput 185:1026–1037. https://doi.org/10.1016/j.amc.2006.07.025

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations