IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 6, 2020, accepted December 20, 2020, date of publication December 29, 2020, date of current version January 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047933

Intrusion Detection System Based on Gradient Corrected Online Sequential Extreme Learning Machine

NEDHAL AHMAD HAMDI QAIWMCHI, HALEH AMINTOOSI, AND AMIRHOSSEIN MOHAJERZADEH
Department of Computer Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad 9177948974, Iran
Corresponding author: Haleh Amintoosi (amintoosi@um.ac.ir)

ABSTRACT Nowadays, Intrusion Detection System (IDS) is an active research topic with machine learning nature. A single-hidden layer feedforward neural network (SLFN) trained on the approach of extreme learning machine (ELM) is used for (IDS). The encouraging factors for its usage are its fast learning and supportability of sequential learning in its online sequential extreme learning machine (OSELM) variant. An issue with OSELM that has been addressed by researchers is its random weights nature of the input-hidden layer. Most approaches use the concept of metaheuristic optimisation for determining the optimal weights of OSELM and resolve the random weight. However, metaheuristic approaches require many trials to determine the optimal one. Hence, there is concern about the convergence aspect and speed. This article proposes a novel approach for finding the optimal weights of the input-hidden layer. This article presents an approach for an integration between OSELM and back-propagation designated as (OSELM-BP). After integration, BP changes the random weights iteratively and uses an iterated evaluation of the generated error for feedback correction of the weights. The approach is evaluated based on various scenarios of activation functions for OSELM on the one hand and the number of iterations for BP on the other. An extensive evaluation of the approach and comparison with the original OSELM reveal a superiority of OSELM-BP in reaching optimal accuracy with a small number of iterations.

INDEX TERMS Online sequential extreme learning machine (OSELM), intrusion detection system (IDS), back-propagation (BP), activation function.

I. INTRODUCTION

Intrusion detection is the task of observing, analysing and identifying activities aiming to violate a network's security policy. The key success factor for identifying such activities relies on an appropriate monitoring of the network by diagnosing its usage chronically [1]. In the past, organisations used specific authentication policies articulating various levels of accessing. The conventional approach used in the past to prevent suspicious activities depended on an authentication framework giving users restricted network access based on their role. Apparently, such approach does not guarantee full prevention of unauthorised activities, where violating a network's privacy has become more advanced [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

Intrusions take various forms, where those intending to accommodate them have purposes other than damaging a network to affect its performance. It is worth mentioning the types of intrusions that might be encountered in a network. The most common intrusion type is the denial of service (DoS), which aims to influence a network's performance by sending massive amounts of information to such network [3]. Another type is probing, which aims to scan a network by searching for a valid IP address to gather information [4]. The third type is usually called compromising, in which an attacker exploits a weakness in a network to get privileged access to it [5] Additionally, there are types of attacks that rely on predefined malicious software, such as viruses, worms and Trojan horses [6]. A common feature of the aforementioned intrusion types is the significant changes that might occur in the network's usage [3], [6] Therefore, the information security community has reacted to such attacks by proposing

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

proper methods that can sniff network packets to report an extensive analysis for what is running on the network. In the literature, the analysis has been divided into two main categories: misused-based analysis and anomaly-based analysis [7].

With the emergence of availably public datasets for intrusions, such as KDD-CUP99 [8] and NSL-KDD [8], the research community has tended to use machine learning techniques (MLTs) for the detection task. Such techniques significantly rely on historical data to build a model that can learn the features of both legitimate and intrusion connections. The model will then be used in future detections. Yet a great debate has been depicted in the literature regarding using MLTs, where the aim was to determine best practice technique. The criteria that have been examined in such debate were related to the training time for building a model and its detection accuracy. Recently, deep learning techniques (DLTs) have caught several researchers' attention due to their significant classification accuracy [9], [10]. This is because most DLTs are based on multilayer neural network architectures that provide better learning and understanding for the intrusion features. However, these architectures are proved to be time-consuming due to longer training times. In this regard, some researchers have attempted to use single-layer neural network (NN) architectures or so-called shallow NN that have a relatively similar classification accuracy but with notably less time consumption. Back-propagation (BP) is one of the most famous approaches for training neural networks and uses the gradient of the error for updating the weights of the neural network until reaching the point of zero gradient, which makes the weights unchanged and the network fully trained. However, such approach was criticised for being slow and subject to local minima by some researchers, which has motivated other competitive approaches that work for shallow networks, such as ELM, which uses the concept of Moore-Penrose inverse to apply the least square error for training in one iteration [11].

The argument for ELM's superiority over BP in terms of accuracy for shallow networks is an open research problem. While many researchers have criticised BP for its low training speed and possibility of falling in local minima, ELM has been criticised for its non-optimal weights in the input-hidden layer because of the random initialisation of the weights as well as the need to define the network's optimal structure [12]. This criticism implies that randomisation obstructs high classification accuracy. In this regard, this paper aims to overcome this drawback by modifying the OSELM using BP to update the input-hidden weights while preserving the OSELM of inheriting hidden-output weights, which would theoretically improve classification accuracy in a short time.

The main contributions of our work are as follows:

1- We propose an integrated OSELM-BP method for IDS to overcome randomisation in input-hidden weights, from which OSELM suffers.
2- The proposed method uses the BP to update the input-hidden weights while preserving the OSELM of inheriting hidden-output weights.
3- We evaluate the performance of the proposed method OSELM-BP in terms of five activation functions, with and without relying on the characterisation model for setting the number of neurons and based on various numbers of iterations added to the BP, and we show its superiority in terms of reaching optimal performance more frequently than OSELM alone.
4- Three datasets are used for evaluation related to IDS, namely, CICIDS-2017, KDD 99 and NSL-KDD 99.

II. LITERATURE SURVEY

Although there are many approaches for intrusion detection in network traffic, such as clustering-based techniques or support vector machines (SVM), they mostly have the disadvantage of long training times. Moreover, they normally need parameter tuning and [13]–[15] and [16], and do not have a satisfactory performance in multiclass classification. Hence, in this section, we focus on machine learning based intrusion detection techniques. The authors in [17] concentrate on ELM and OSELM techniques used for the IDSs. These methods have several attributes that motivate the usage to build IDSs, including (i) easy assignment of parameters, (ii) perfect generalisation and (iii) online and fast training. The results indicate that the methods can be simply used for a great amount of data without considerable loss of generalisation. In [18], an OSELM-based technique is provided for intrusion detection. The proposed technique uses the profiling of alpha for reducing time complexity when the irrelevant attributes are discarded by using correlation, consistency, filtered ensemble-based techniques for attribute selection. Instead of sampling, beta profiling is used for reducing the training dataset size. For the performance evaluation of the proposed technique, a standard NSL-KDD 2009 dataset is used. The authors in [19] provide the technique for intrusion detection based on OSELM. For the performance evaluation, a KDD-CUP99 dataset is used. In the study, they use three subset evaluations of attribute selection techniques: filtered evaluation, CFS subset evaluation, and consistency subset evaluation for removing redundant attributes. Two techniques of network traffic profiling are used. Alpha profiling is performed to reduce time complexity, and beta profiling is used to remove redundant connection records, thus decreasing dataset size. In [20], the OSELM-based intrusion detection system appeared and was used to detect attacks in advanced metering infrastructure (AMI) and perform comparative analyses on other algorithms. The results of the simulation indicate that, compared with other methods of intrusion detection, the method of OSELM-based intrusion detection is better in terms of detection speed and accuracy. In [21], the new DA-ROS-ELM (dual adaptive regularised online sequential extreme learning machine) is provided for detecting network intrusion. The Tikhonov regularisation-based ridge regression factor is defined for solving problems that are ill-posed and over-fitting. For the arrived data in every step of updating, as well as the whole recently accessible data, the mechanism of dual adaptive is planned for, respectively, the selection of accurate updating of output weight β, as well as the regularised parameter C. The proposed algorithm performance is evaluated using the NSL-KDD dataset. The results indicate that DA-ROS-ELM can achieve greater generalisation and performance, higher accuracy, lower rates of false positives and false negatives and faster speeds of training than other network intrusion detection algorithms. In [22], as in batch ELM, in OSELM-RLS single-hidden layer feedforward neural network (SLFN) input weights are produced randomly, although output weights are gained by the solution of recursive least-squares RLS. In [23], they provide the OSELM using the efficient mechanism of sample updating. Old and novel samples are considered various weights. The effect of novel training samples on the algorithm is increased further, which is able to further promote ELM regression prediction ability. Simultaneously, the improved algorithm of the artificial bee colony is provided and used for optimising an adaptive OSELM parameter. A proposed prediction method stability and a convergence property are proved. Real gathered short-term wind-speed time series are used as objects of research and confirm proposed method prediction performance. Short-term wind-speed multi-phase prediction simulation is done. In comparison to the other methods of prediction, the results of the simulation indicate that the proposed approach has reliability performance, a higher accuracy of prediction and increased indicators of performance. In the work of [20], a simple OSELM-based IDS system was applied to detect intrusion attacks in a smart grid. The authors have not tackled the issue of random weights between the input and hidden layer. In the work of [24], an ensemble of OS-ELM machine (EOSELM) feature selection was proposed to predict the post-fault transient stability status of power systems in real time. An integration of OSELM as a weak classifier and an online boosting algorithm as an ensemble learning algorithm was done. In the work of [25], a distribution of the existing centralized cloud intelligence is done to local fog nodes to detect the attack at faster rate for IoT application where online sequential extreme learning machine (OS-ELM) was used for this purpose.

Some researchers have focused on the implementation aspect of ELM for IDS in fog networks. For example, in the work of [26], a distributed ELM classification for fog networks is proposed, in which each node of the fog is trained on the sample of the entire data considering that this sample represents accessible data by the node in the fog. The authors have derived a classical ELM model by indexing it according to the node of the fog, and requesting that the training process is repeated until reaching a minimum needed performance of training error, which the authors called a performance index. This work also suffers from the issue of random weights of ELM. The use of ELM for IDS was also used with probabilistic algorithms. This is shown in the work of [1], where a probability density function is learned based on flow features for frequent communications. The authors have used a hierarchical heavy hitters' algorithm for clustering network statistics and learning the probability density function of each feature using ELM. Moreover, this model has not dealt with the random weights of ELM. Some researchers have integrated ELM with feature reduction algorithms, namely principle component analysis (PCA) for boosting performance and reducing computational time. This is done in the work of [27], where an adaptive PCA was used with ELM. However, this is regarded as a direct implementation of ELM without any handling of the random weights issue of ELM. Other proposed methods of using ELM for IDS were by proposing various architectures of classification. In the work of [28], a cascade architecture based on a set of ELM individual classifiers was proposed to counter the issue of imbalance of an IDS dataset due to the majority being normal samples and the minority being attack samples.

Meta-heuristic based ELM optimization was also used extensively. The work of [29], where a particle swarm optimisation (PSO) was used to maximise an objective function representing the training accuracy of the network based on a solution space. The solution space contains the candidate weights of the connections between the input and hidden layers and the biases of the hidden neurons. This approach provides better accuracy than an arbitrary weight of NN in the input-hidden layer. However, there is a concern about computational complexity due to the need of considerable searching based on an adequate number of particles and iterations before reaching a convergence state. The literature contains numerous attempts of optimising the weights of the neural network in ELM using metaheuristic optimisation algorithms, such as differential evolution [30], [31], cuckoo search [32], the firefly algorithm [33], dolphin swarm optimisation [34], genetic optimisation [35], and ameliorated teaching-learning-based optimization [36]. Additionally, an attempt to optimise the number of hidden neurons in ELM is the work of [37], where a greedy approach was proposed between a candidate minimum number and maximum number, and the training error was used as a metric to select the number with the best performance. Obviously, such work is subject to local minima, as the performance is not necessarily a convex function with respect to the number of hidden neurons. Hence, another attempt to optimise the number of hidden neurons was done based on a metaheuristic approach instead of greedy searching, as in the work of [38].

Overall, ELM has been used for IDS in both its offline and online learning mode. The lightweight nature of this model makes it appealing to be deployed in IDS. However, researchers aim at improving the accuracy of both ELM and OSELM when it is used for IDS to reduce the number of false alarms. This has motivated researchers to focus on the issue of random weights in the input hidden layer in the model. Most studies have concentrated on using metaheuristic searching for this purpose, which leads to optimal weight. However, there is concern about the convergence performance of the

TABLE 1. The symbols that are used in the article.

Symbol	Meaning
(x_j, t_j)	Data decomposed into feature and targets
$\{X_0, Y_0\}$	Boosting data
$w_i = (a_i, b_i)$	Input hidden weights
β_i	The output weights
$g()$	The activation functions
H	The hidden output matrix
C	Regularization factor
L	Number of hidden neurons
$k()$	Kernel
η	Learning rate
$MaxIt$	The maximum number of iterations for BP
Δw_i	The changing in the weights in one iteration
N	The size of the data
d	The number of features
m	The number of outputs
$\beta^{(0)}$	The weights of the hidden-output layer after training on boosting data
M_0	The inverse of $H_0^T Y_0$ and it is used as intermediate matrix for sequential training
$\beta^{(k)}$	The weights of the hidden-output layer after training on chunk (k)

searching when using metaheuristic searching. Observing the essence in the difference between ELM training and BP training is in the gradient of error usage that gives BP the ability to converge gradually towards the minimum error point. However, the ELM approach lacks this behavior due to the one-shot calculation using the concept of least square error. It would be interesting if a novel approach was proposed with leveraging the advantages of each of them. The lightweight, over-fitting and local optimal avoidance nature of ELM and the gradual convergence to the optimal point of BP are the goals of the article.

III. METHODOLOGY

This section presents the methodology of the integrated OSELM-BP learning. It starts with an overview of the classical OSELM model in sub-section III.A. Next, we present an overview of back-propagation in sub-section III.B. Next, we present our integrated OSELM-BP in sub-section III.C. The computational complexity is discussed in Section III.D. The datasets that are used for evaluation are provided in sub-section III.E, and the evaluation metrics are presented in sub-section III.F. Table 1 demonstrates the notations used.

A. MODEL FORMULATION

Assuming that we have N arbitrary distinct samples $(x_j, t_j) \in \mathcal{R}^d \times \mathcal{R}^m$ where d denotes the number of features (attributes) and m denotes the number of outputs (targets). In addition, we assume that we have single layer feed-forward neural network (SLFN) combined of L hidden neurons then, we approximate the weights using the model in (1):

$$\sum\nolimits_{i=1}^{L} \beta_i g\left(a_i, b_i, x_j\right) = t_j \tag{1}$$

where

i denotes the index of number of neuron number $i = 1, ..L$

j denotes the index of sample $j = 1, 2 \ldots N$

Another compact way to write the previous equation is (2):

$$H\beta = T \tag{2}$$

where

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} g(a_1, b_1, x_1) & \ldots & g(a_L, b_L, x_1) \\ & \vdots & \\ g(a_1, b_1, x_N) & \ldots & g(a_L, b_L, x_N) \end{bmatrix}$$

H is combined of N rows and L columns

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix}$$

$$T = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}$$

The constraint of number of columns of H (or L) being equal to the number of rows of β is applied which makes the equation valid. The way of training that Huang has suggested in his article of ELM starts with random initialization of $(a_i, b_i) \in [-1, 1]$ and then finding the value of β based on (3):

$$\beta = H^{-1}T \tag{3}$$

Unfortunately, H is not guaranteed to be square because $N \gg L$. Hence, we perform the Moore-Penrose generalized inverse of matrix $H^{\dagger} = \left(H^T H\right)^{-1} H^T$. Hence, the total equation of $\beta = \left(H^T H\right)^{-1} H^T T$. Some researchers have suggested adding a positive value $\frac{1}{C}$ in the equation in order to make the solution more stable and to have more generalization according to ridge regression theory. It is named as regularization factor, as shown in (4):

$$\beta = (\frac{1}{C} + H^T H)^{-1} H^T T \tag{4}$$

TABLE 2. **Pseudocode of integrated BP-ELM training.**

```
Input

$(x_j, t_j)$ Training Data
$j = 1, \dots N$
$g$ activation function
$It$
Output
$(a_i, b_i, \beta_i), i = 1, \dots L$


Start
    1-   Generate random values for $(a_i, b_i) \in [-1,1]$
    2-   Calculate the matrix $H$ based on $x_j$ and $g$
    3-   Apply (4) for calculating $\beta_i$

    4-   Calculate $C(y(x_j), t_j)$
    5-   For iterations starting from 1 until $MaxIt$
             Find the derivative $\Delta w_i$
             Update the weights using (7)
             Calculate $C(y(x_j), t_j)$

    6-   End
```

After training, the prediction of any new sample will be based on (5):

$$y = f\,(x) = h(x)(\frac{1}{C} + H^T H)^{-1} H^T T$$
$$= h(x) H^T (\frac{1}{C} + HH^T)^{-1} T \qquad (5)$$

More generalization of the equation is done by using kernel variant in the form of (6):

$$y = \begin{bmatrix} k\,(x, x_1) \\ \vdots \\ k\,(x, x_N) \end{bmatrix} (\frac{1}{C} + \Omega_{ELM})^{-1} T \qquad (6)$$

Ω_{ELM} denotes the kernel matrix

k denotes the kernel of the new sample with respect to the training data.

Regardless of the variant that is used for ELM, there is an issue in the weights of the input hidden layer w_{ij}=(a_{ij}, b_{ij}) which is the random generation. Such random generation causes non-stable performance as well as sub-optimal solutions. For solving this, the concept of back-propagation is adopted. Assuming that the C is a loss function (or cost function) which measures the number of mis-classifications in the neural network after being trained with traditional ELM. In addition, we assume that the overall network is given as $y(t_j) = h(x_j)H^T(\frac{1}{C} + HH^T)^{-1} T$ for training set combined of pairs j, (x_j, t_j). Hence, the loss of the model on that pair (x_{new}, y_{new}) is the cost of the difference between the output $y\,(x_j)$ and the target t_j is $C(y\,(x_j)\,, t_j)$. We consider that the cost is represented by error or misclassification E. We calculate the derivative of the error with respect to the weights, we change the weights using (7):

$$\Delta w_i = -\eta \partial \frac{E}{w_i} \qquad (7)$$

where

η denotes the learning rate

$$w_i{=}(a_i, b_i)$$

We conduct set of iterations $MaxIt$ where is each one the value w_i is updated using (8):

$$w_i^{new} = w_i^{old} + \Delta w_i \qquad (8)$$

B. ONLINE SEQUENTIAL EXTREME LEARNING MACHINE OSELM

In our algorithm, we adopt an online variant of extreme learning machine called single hidden layer feed-forward [39]. The online variant enables the updating of the knowledge of the NN according to the provided chunks. Given an activation function g and L hidden neurons, the learning procedure consists of two phases described below.

1) BOOSTING PHASE USING THE INITIAL CHUNKS

Given a small initial training set $\{X_0, Y_0\}$ to boost the learning algorithm first through the following boosting procedure.

1) initialize arbitrary input weight w_i and bias b_i based on a random variable with center u_i and standard deviation σ_i, $i = 1, \dots, .L$.
2) Calculate the initially hidden layer output matrix in (9).

$$\mathrm{H_0} = [\mathrm{h_1}, \dots, \mathrm{H_L}]^{\mathrm{T}} \qquad (9)$$

where, $h_i = [g\,(w_i \cdot x_i + b_i)\,, \dots, g\,(w_L \cdot x_i + b_L)]^T$ and, $i = 1, \dots, L$.

3) Estimate the initial output weight (10).

$$\beta^{(0)} = \mathrm{M_0 H_0^T Y_0} \qquad (10)$$

where, $\mathrm{M_0{=}(H_0^T Y_0)^{-1}}$ and $\mathrm{Y_{0{=}}[y_{1,\dots,}y_L]^T}$ Set $K = 0$.

TABLE 3. Pseudocode of integrated OSELM-BP training.

```
Input
$\{X_0, Y_0\}$ boosting data
$(x_j, t_j)$ Training Data
$j = 1, \dots N$
$g$ activation function
$MaxIt$
$\eta$ learning rate

Output
$(a_i, b_i, \beta_i), i = 1, \dots L$

Start
Generate random values for $(a_i, b_i) \in [-1,1]$
Calculate the matrix $H_0$ based on $X_0$ and $g$
Apply (9) for calculating $\beta^{(0)}$
For each chunk (k)
Update $\beta^{(k)}$ to $\beta^{(k+1)}$ using (11), (12)
  Calculate $C(y(x_j), t_j)$
  For iterations starting from 1 until $MaxIt$
      Find the derivative $\Delta w_i$
      Update the weights using (6), (7)
      Calculate $C(y(x_j), t_j)$
   End
End
```

2) SEQUENTIAL LEARNING PHASE

For each further coming observation (X_i, Y_i), where, $X_i \in i = L+1, L+2, L+3, \dots,$

1) Calculate the hidden layer output vector by using (11).

$$h_{(k+1)} = [g(w_1 \cdot x_i + b_1), \dots, g(w_L \cdot x_i + b_L)]^T \quad (11)$$

2) Calculate the latest output weight $\beta^{(k+1)}$ based on Recursive Least Square (RLS) algorithm shown in (8) to (12).

$$M_{k+1} = M_k - \frac{M_k h_{k+1} h_{k+1}^T M_k}{1 + h_{k+1}^T M_k h_{k+1}} \quad (12)$$

$$\beta^{(k+1)} = \beta^{(k)} + M_{k+1} h_{k+1} (y_i^T - h_{k+1}^T \beta^{(k)}) \quad (13)$$

Set $k = k + 1$

C. INTEGRATED BACK-PROPAGATION ONLINE SEQUENTIAL EXTREME LEARNING MACHINE- OSELM-BP

This section presents the integrated back-propagation OSELM (OSELM-BP). It is combined of both boosting phase and iterative phase similar to OSELM. A general flowchart for the algorithm is presented in Fig. 1. As it is depicted in the pseudocode, the algorithm uses the boosting data $\{X_0, Y_0\}$, the chunks (x_j, t_j), the activation function g, the number of iterations $MaxIt$, and the learning rate η. The algorithm basically performs boosting for the neural network in the initial stage, and then it updates the weights iteratively with new chunk by calling both OSELM for initial update of the weights and calling back-propagation for iterative update of the weights using the factor η for $MaxIt$.

D. COMPUTATIONAL COMPLEXITY

The computational complexity of OSELM-BP spends more time on back-propagation training after the phase of ELM. The extra time is only $\mathcal{O}(mLMaxIt)$ where m denotes the number of output neurons, L denotes the number of hidden neurons and $MaxIt$ denotes the number of iterations. Considering that $MaxIt$ was set to not exceed 200 this makes and the number of output neurons equals to the number of classes which is still low number in IDS problems, the added computational complexity does not affect the practical applications

E. DATASETS

This section presents example of the most famous datasets in IDS. We present each one with providing its statistical information from the perspective of number of records, classes and their decomposition.

1) CICIDS-2017

CICIDS-2017 dataset [40]. The details of the dataset are provided in Table 4. It shows the name of the used file, the day of activity and the found attack.

In our experiments, we merge all the traffic data within the five days (as shown in Table 4) in a single dataset. Table 5 depicts the details of the merged dataset.

2) KDD 99

Since 1999, KDD'99 has been the most wildly used data set for the evaluation of anomaly detection methods [41]. This data set is built based on the data captured in DARPA'98 IDS evaluation program DARPA'98 is about 4 gigabytes of compressed raw (binary) TCP dump data of 7 weeks of network traffic, which can be processed into about 5 million

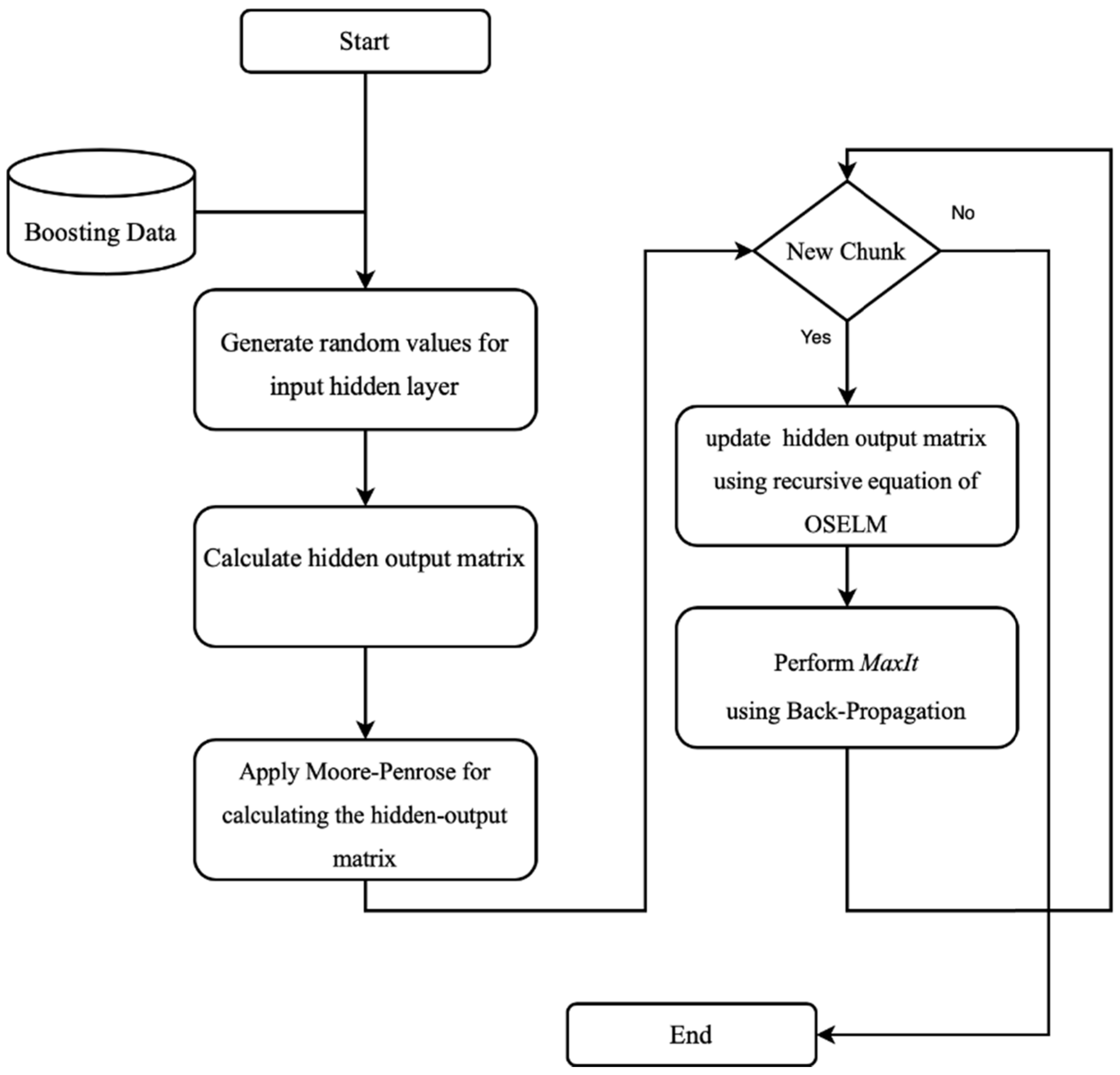
Start
Boosting Data
Generate random values for input hidden layer
Calculate hidden output matrix
Apply Moore-Penrose for calculating the hidden-output matrix
New Chunk
No
Yes
update hidden output matrix using recursive equation of OSELM
Perform *MaxIt* using Back-Propagation
End

FIGURE 1. **Flowchart of integrated online sequential extreme learning machine and back propagation OSELM-BP.**

connection records, each with about 100 bytes. The two weeks of test data have around 2 million connection records. KDD 99 training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type. The simulated attacks fall in one of the following four categories:

Denial of Service Attack (DoS): is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users' access to a machine.

User to Root Attack (U2R): is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.

Remote to Local Attack (R2L): occurs when an attacker who has the ability to send packets to a machine over a network but who do not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

Probing Attack: is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

The distribution of the classes according to the sample's sizes are provided in pie graph in Fig. 2. As we observe, there is an unbalance in the dataset. This un-balance makes the problem of classification or clustering very challenging.

3) NSL-KDD

The statistical analysis showed that there are important issues in the data set which highly affects the performance of the systems, and results in very poor estimation of anomaly detection approaches. To solve these issues, a new data set as, NSL-KDD is proposed, which consists of selected records of the complete KDD 99 data set [42], [43]. The advantage of NSL KDD dataset is.

TABLE 4. **Dataset description.**

Name of files	Day of activity	Attacks found
Monday-WorkingHours.pcap_ISCX.csv	Monday	Benign (Normal human activities)
Tuesday-WorkingHours.pcap_ISCX.csv	Tuesday	Benign, FTP-Patator, SSH-Patator
Wednesday- WorkingHours.pcap_ISCX.csv	Wednesday	Benign DoSGoldenEye, DoS Hulk, DoSSlowhttptest, DoSslowloris, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Thursday	Benign, Web Attack-Brute Force, Web Attack-Sql Injection, Web Attack-XSS
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv	Thursday	Benign Infilteration
Friday-WordkingHours-Morning.pcap_ISCX.csv	Friday	Benign, Bot
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Friday	Benign, PortScan
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Friday	Benign, DDoS

TABLE 5. **General description of CICIDS-2017 dataset.**

Attribute	Description
Total number of distinct instances	2830540
Number of features	83
Number of distinct classes	15

No redundant records in the train set, so the classifier will not produce any biased result.

No duplicate record in the test set which has better reduction rates.

The number of selected records from each difficult level group is inversely proportional to the percentage of records in the original KDD 99 data set.

The training dataset is made up of 21 different attacks out of the 37 presents in the test dataset. The known attack types are those present in the training dataset while the novel attacks are the additional attacks in the test dataset i.e. not available in the training datasets.

F. EVALUATION METRICS

This section presents the evaluation measures used for quantifying the performance of the proposed OSELM-BP and the comparison with OSELM. TP denotes true positive, TN denotes true negative, FP denotes false positive, and FN denotes false negative.

1) ACCURACY

Accuracy represents the number of true predictions divided by all cases of prediction [24], The formula is calculated in (14).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

2) PRECISION (PPV)

Positive predictive value (PPV) represents the number of true positive predicted by the classifier divided by the number of all predicted positive records [25], The formula is calculated in (15).

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{15}$$

3) RECALL (TPR)

TPR represents the number of TP predicted by the classifier divided by the number of all tested positive records [26], The formula is calculated in (16).

$$TPR = \frac{TP}{TP + FN} \tag{16}$$

4) G-MEAN

This measure is calculated based on precision and recall. [25], The formula is calculated in (17).

$$G\_mean = \frac{TP}{\sqrt{(TP + FP)\,(TP + FN)}} \tag{17}$$

5) F-MEASURE

This measure is the harmonic mean of the precision and recall [25]. It is calculated based on the equation; the formula is calculated (18).

$$F\_measure = \frac{2\times precision \times recall}{precision + recall} \tag{18}$$

IV. EXPERIMENTAL DESIGN AND RESULTS

The simulations for OSELM and OSELM-BP algorithms are carried out in the MATLAB 2019b environment running in Intel Core i5 CPU with the speed of 1.4 GHz.

The experimental design starts with building the characterization model which creates the relation between the testing accuracy and the number of neurons in the hidden layer. For each of the three datasets we generate the characterization model that is used for finding the best number of neurons to operate OSELM and OSELM-BP. We find that each characterization model has accomplished a peak at different number of neurons as it is presented in Fig. 2.

The main parameter is the number of neurons and it was determined based on characterization model given in Fig. 2 for each of the three datasets (a) CICIDS2017 dataset (b) KDD99, (c) NSL. The number of neurons is selected from the characterization model given in Fig. 2, which is 100, 300 and 100 for CICIDS2017, KDD99 and NSL, respectively

For evaluation, the data was partitioned into 60 vs. 40 percentages for training and testing respectively. The proposed OSELM-BP will be compared with the original OSELM with respect to six evaluation metrics, namely, accuracy, precision, recall, F-measure, G-mean and the time. The first fifth evaluation metrics are to be maximized while the last one is to be minimized. Thus, we show the reciprocal of the time and we normalize it to one. Each of the two models will be considered for one possible type of five type of activation functions: sigmoid, hardlim, rbf, tensing and sin. In addition, we consider for OSELM-BP one case of four cases of iterations: 9, 24, 99 and 199. The experiments were repeated for two separated sets: the first one is when the number of neurons was taken to be the same of the characterization model while the second set when the number of neurons was as two third of the number of features. Also, each set is repeated for the three datasets: CICIDS-2017, KDD99 and NSL-KDD. We show the testing results of characterization model based selection of number of neurons in Figures 3,17a. Observing the figures, we see that OSELM has behaved better for two activation functions sin and sigmoid while OSELM-BP was better RBF, tansig and hardlim. The poor performance of OSELM-BP in the case of sin and sigmoid is interpreted by over-fitting because the optimal number

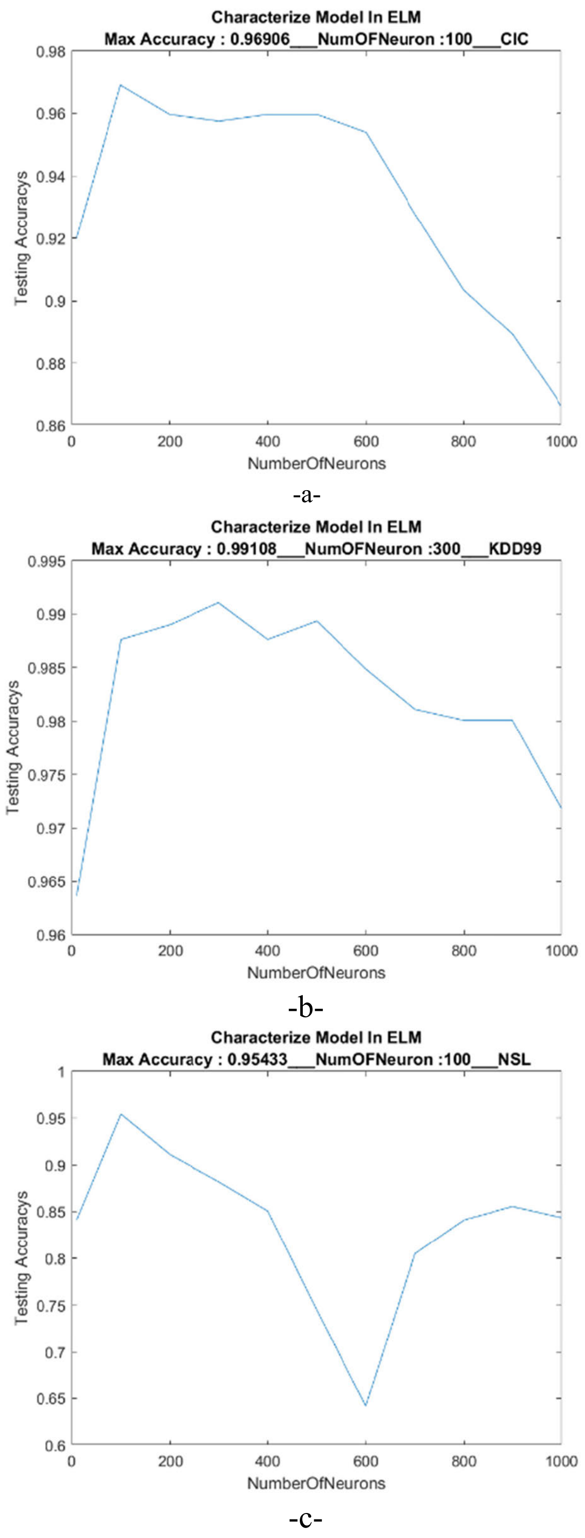

FIGURE 2. Characterization model for validation accuracy and number of neurons -a- CICIDS2017 dataset -b- KDD99 -c-NSL.

of neurons is determined using the characterization model. Thus, adding more iterations of training from BP results in over-fitting and poor performance. Another observation is

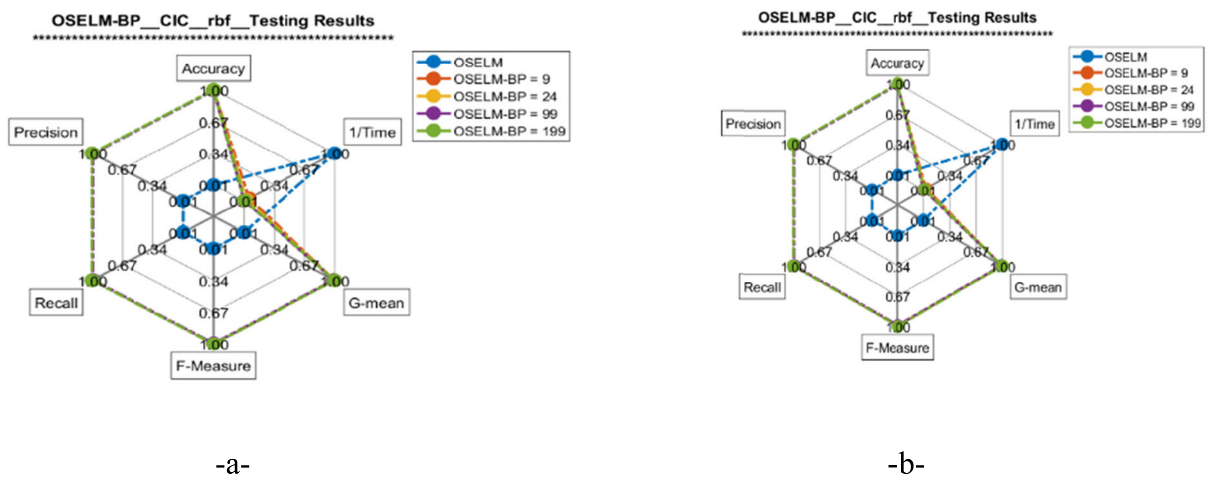

FIGURE 3. The comparison between OSELM and OSELM- BP with respect to iterations 9, 24, 99 and 199 for CICIDS-2017 data set and RBF activation function -a- with characterization model -b-without characterization model.

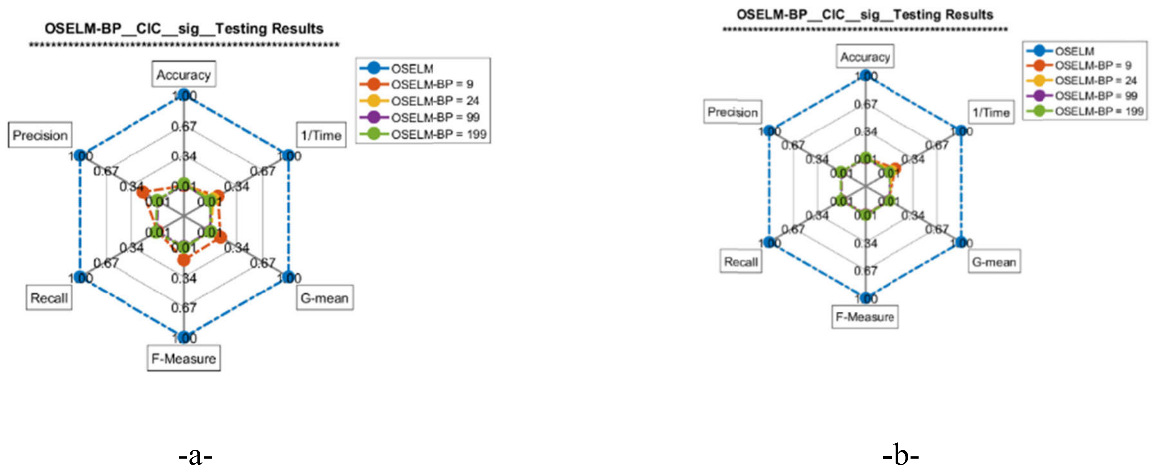

FIGURE 4. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for CICIDS-2017 data set and sig activation function -a- with characterization model -b-without characterization model.

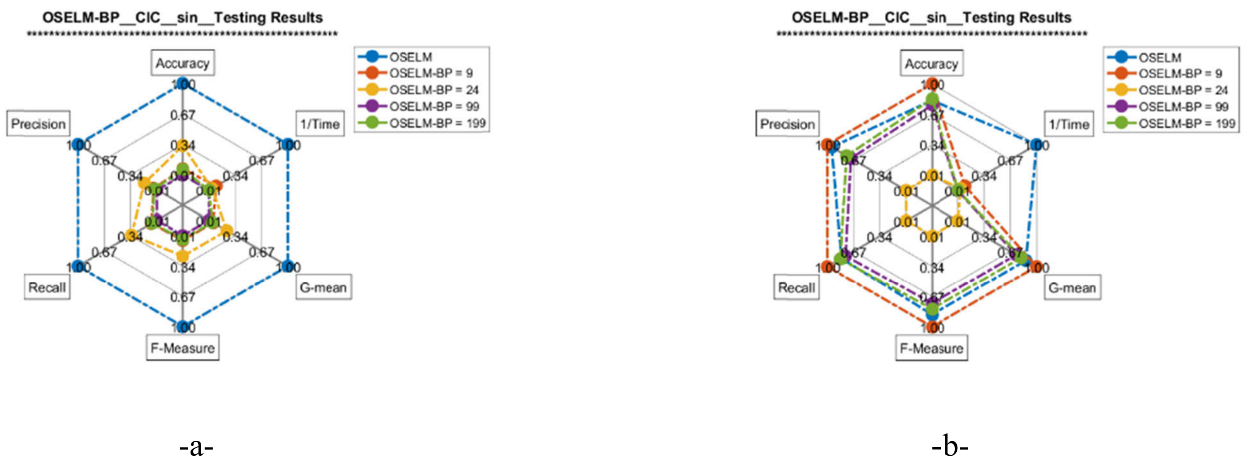

FIGURE 5. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for CICIDS-2017 data set and sin activation function -a- with characterization model -b-without characterization model.

that OSELM-BP has accomplished better accuracy, precision, recall, F-measure and G-mean for RBF activation function in CICIDS-2017 dataset which indicates to the role of BP iterations for improving the performance of classification even if the number of neurons was selected based on characterization model. Furthermore, we observe that

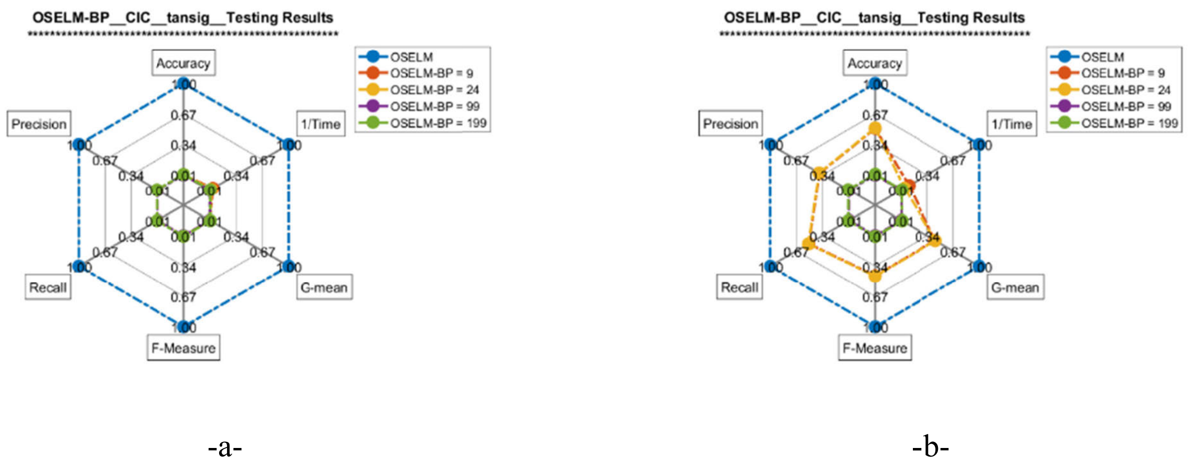

FIGURE 6. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for CICIDS-2017 data set and tansig activation function -a- with characterization model -b-without characterization model.

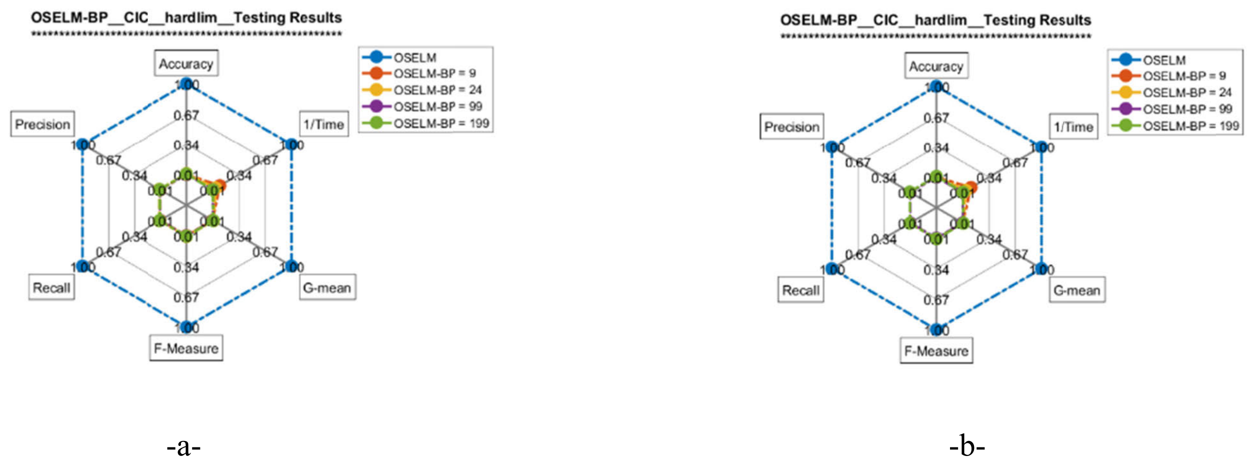

FIGURE 7. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for CICIDS-2017 data set and hardlim activation function -a- with characterization model -b-without characterization model.

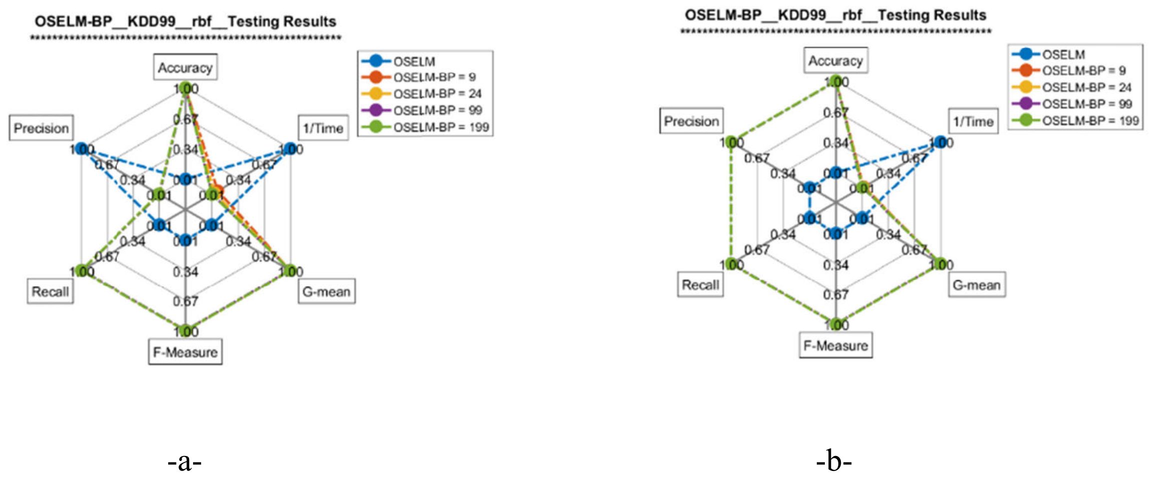

FIGURE 8. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for KDD99 data set and RBF activation function -a- with characterization model -b-without characterization model.

in Fig. 16a, 10 iterations for OSELM-BP were adequate to bring the model to the highest performance while when the number of iterations increases the performance has declined, however, in all cases OSELM-BP was superior over OSELM. Another observation is that the model of OSELM-BP is dependent on the number of iterations for reaching the

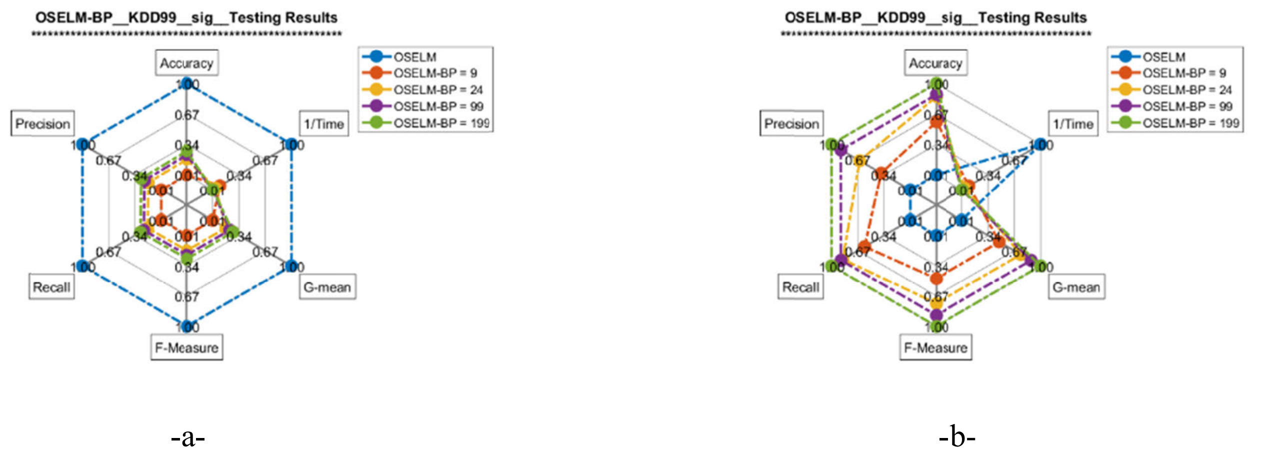

FIGURE 9. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for KDD99 data set and sig activation function -a- with characterization model -b-without characterization model.

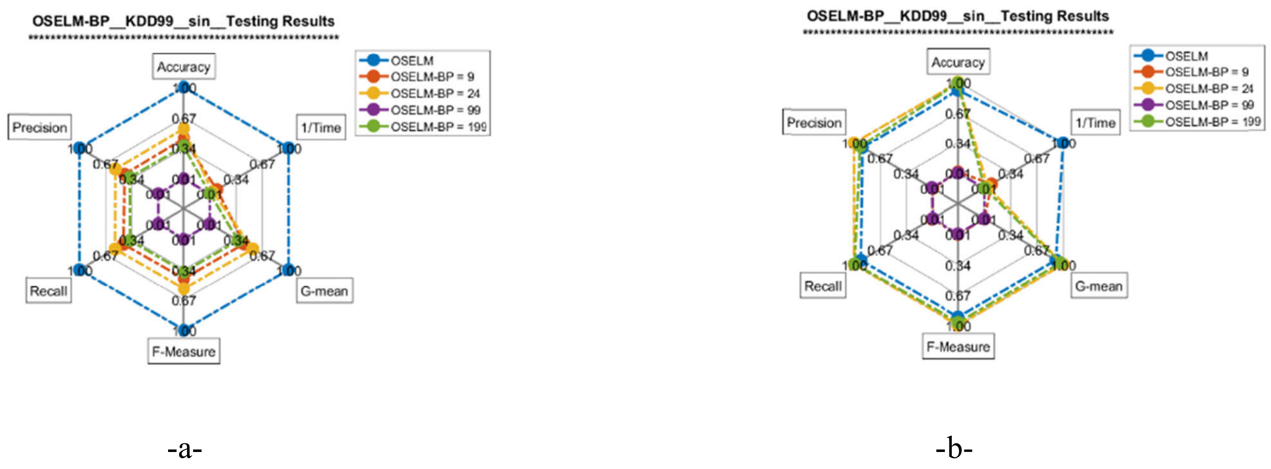

FIGURE 10. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for KDD99 data set and sin activation function -a- with characterization model -b-without characterization model.

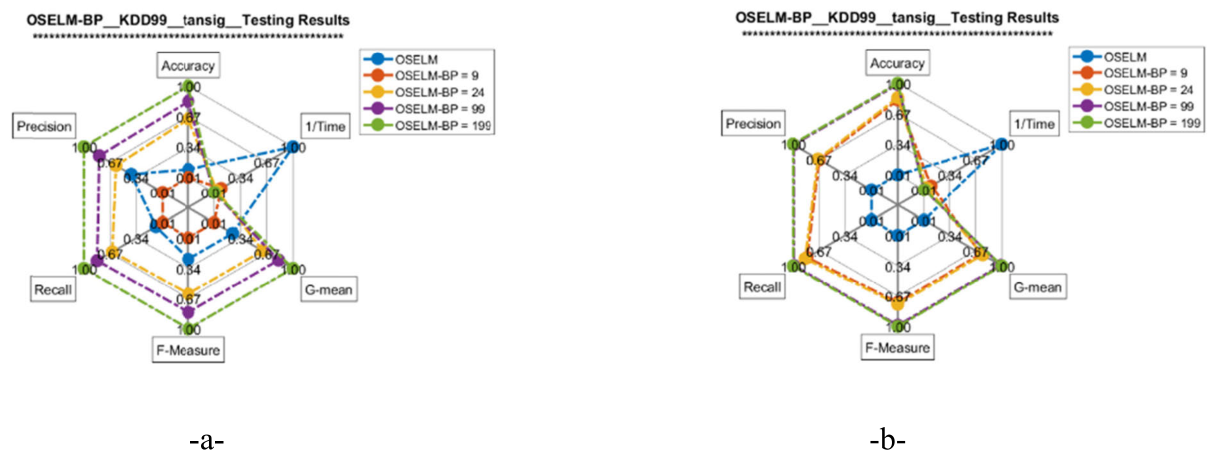

FIGURE 11. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for KDD99 data set and tansig activation function -a- with characterization model -b-without characterization model.

optimal performance. For example, the optimal performance was reached at number of iterations 9 for NSL-KDD data set and hardlim activation function in Fig. 16.a while it has reached it at number of iterations 199 for NSL-KDD data set and tansig activation function in Fig. 15.a. Also, we observe from figures that in all datasets when activation functions

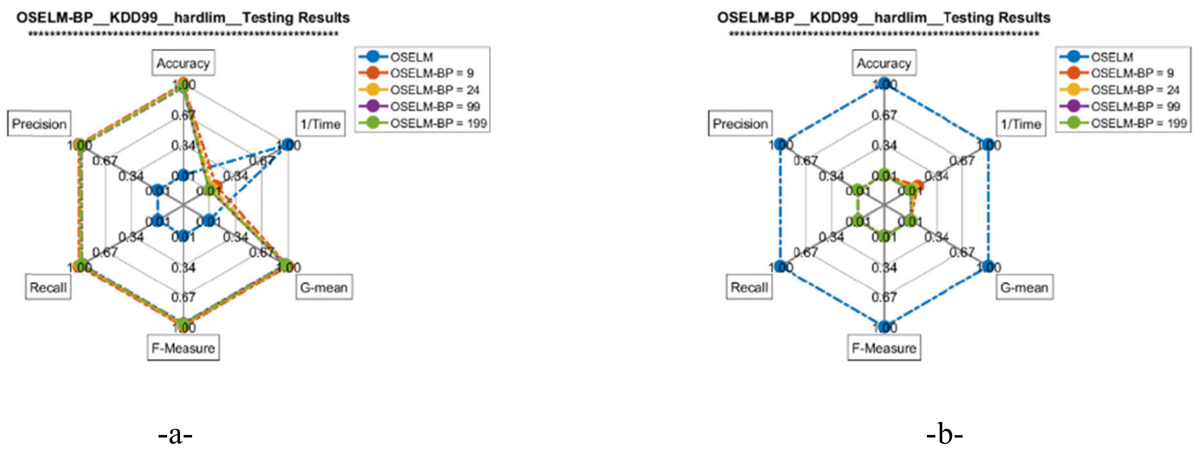

FIGURE 12. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for KDD99 data set and hardlim activation function -a- with characterization model -b-without characterization model.

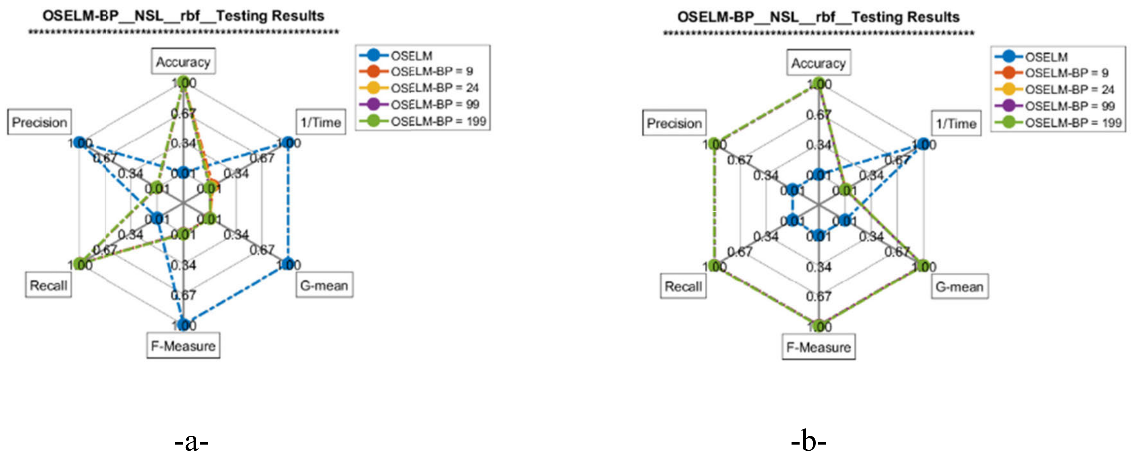

FIGURE 13. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for NSL-KDD data set and rbf activation function -a- with characterization model -b-without characterization model.

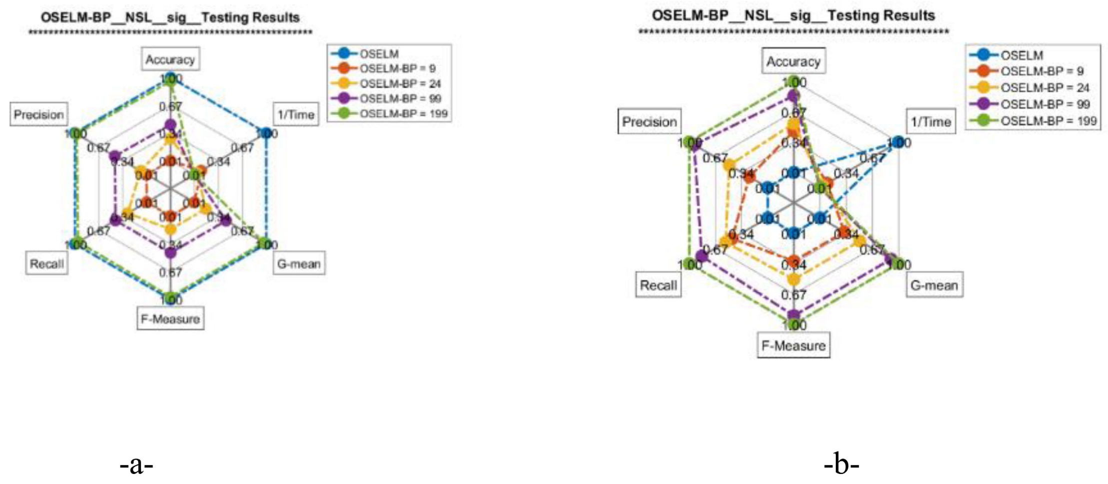

FIGURE 14. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for NSL-KDD data set and sig activation function -a- with characterization model -b-without characterization model.

RBF, sin and tansig were used, OSELM-BP has outperformed OSELM when the number of neurons was selected as two third of the number features. Hence, OSELM-BP is capable of brining the model to highest classification performance with the least possible number of neurons. Contrary to OSELM which has generated less performance when the

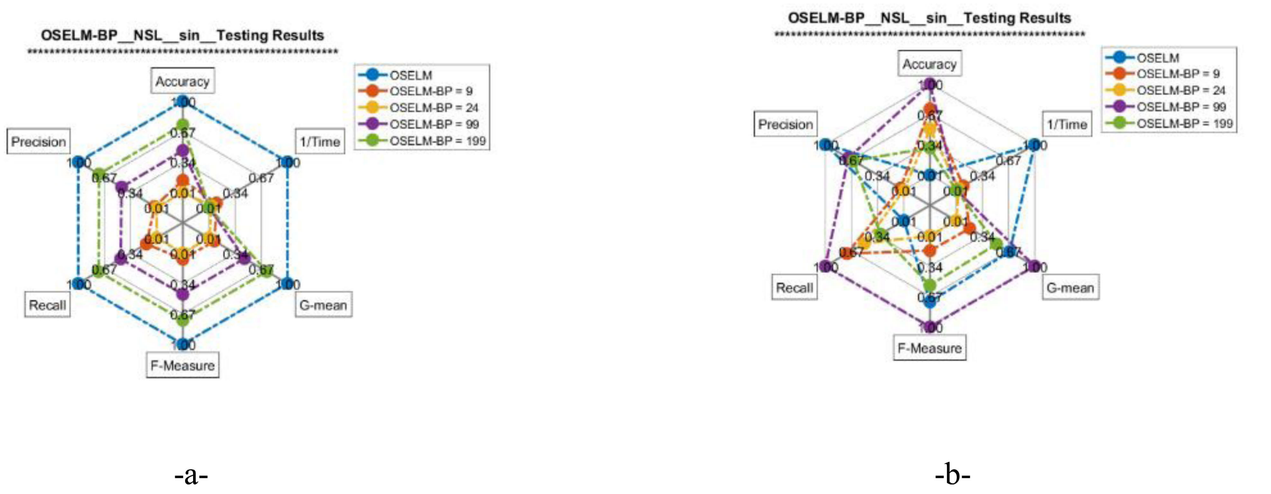

FIGURE 15. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for NSL-KDD data set and sin activation function -a- with characterization model -b-without characterization model.

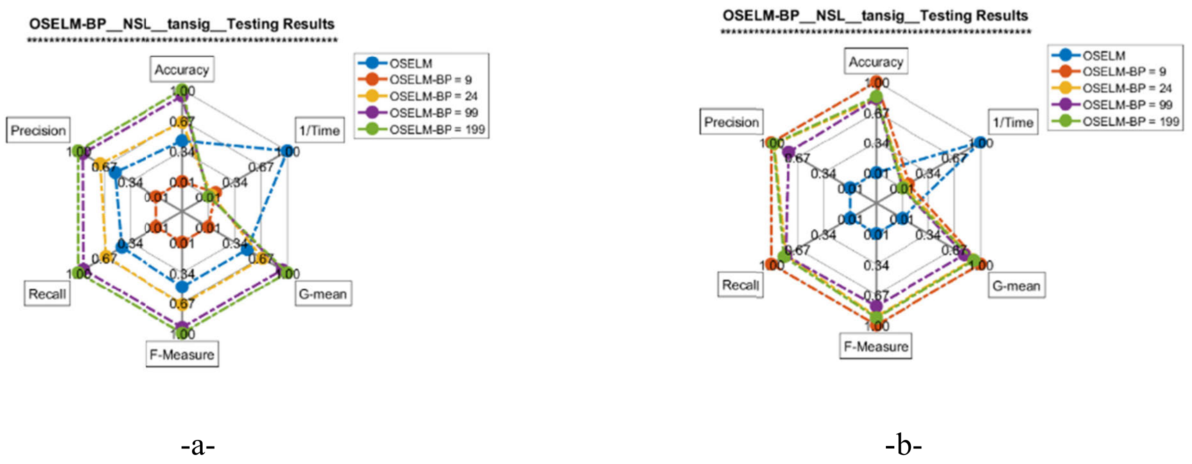

FIGURE 16. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for NSL-KDD data set and tansig activation function -a- with characterization model -b-without characterization model.

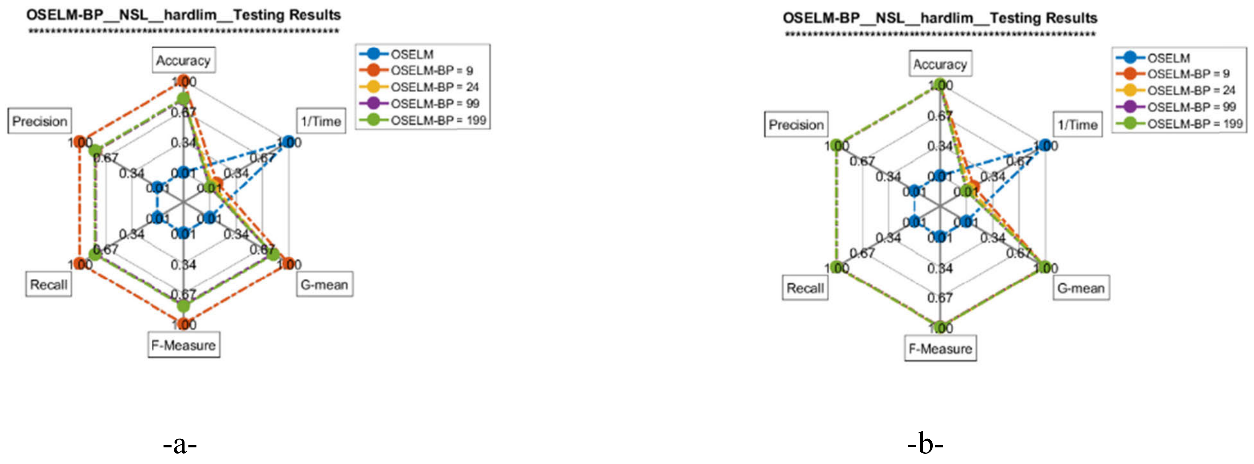

FIGURE 17. The comparison between OSELM and OSELM-BP with respect to iterations 9, 24, 99 and 199 for NSL-KDD data set and hardlim activation function -a- with characterization model -b-without characterization model.

number of neurons was selected minimum without depending on the characterization model.

In order to summarize the superiority between OSELM and OSELM-BP, we present Tables 6 and 7. The results in the tables reveals that when the characterization model OSELM-BP was superior over OSELM in 8 cases out of 15 while the latter was superior in 7 only. On the other side, when the characterization model was not used in

TABLE 6. Summary of superiority between OSELM and OSELM-BP for different datasets and activation function without using characterization model.

With characterization	RBF	Sig	sin#	tansig	hardlim
CICIDS-2017	OSELM-BP	OSELM	OSELM	OSELM-BP	OSELM
KDD99	OSELM-BP	OSELM	OSELM	OSELM-BP	OSELM-BP
NSL-KDD	OSELM-BP	OSELM	OSELM	OSELM-BP	OSELM-BP

TABLE 7. Summary of superiority between OSELM and OSELM-BP for different datasets and activation function when using characterization model.

With characterization	RBF	Sig	sin#	tansig	hardlim
CICIDS-2017	OSELM-BP	OSELM	OSELM	OSELM-BP	OSELM
KDD99	OSELM-BP	OSELM	OSELM	OSELM-BP	OSELM-BP
NSL-KDD	OSELM-BP	OSELM	OSELM	OSELM-BP	OSELM-BP

TABLE 8. Numeric representation of best achieved accuracy for the models.

Dataset-Method	accuracy	precision	recall	F-measure	G-mean
CIC-OSELM	0.9867	0.9613	0.9600	0.9607	0.9607
CIC-OSELM-BP	**0.9873**	**0.9633**	**0.9620**	**0.9626**	**0.9626**
KDD99-OSELM	0.9914	0.9827	0.9786	0.9806	0.9807
KDD99-OSELM-BP	**0.9958**	**0.9893**	**0.9896**	**0.9895**	**0.9895**
NSL-OSELM	0.9973	0.9676	0.9713	0.9695	0.9695
NSL-OSELM-BP	**0.9989**	**0.9859**	**0.9889**	**0.9874**	**0.9874**

selecting the number of neurons in the hidden layer, OSELM-BP was superior in 11 cases and equivalent in one case while OSELM was only superior in 3 cases. Such results support the hypothesis of the effectiveness in using back-propagation with OSLEM for fine tuning of the model after OSELM is performed. The interpretation of lacking the superiority in some cases arises because of the over-fitting that occurs in some cases. Obviously, the number of over-fitting is more when the characterization model is used because the suitable number of neurons in the hidden layer was determined prior to the training.

Based on the above analysis, it can be stated that OSELM-BP has high scalability. The evaluation reveals the superiority of OSELM-BP in reaching optimal accuracy with a small number of iterations even with large data like KDD 99 which makes it high scalable approach. For overall summary of the performance differences between OSLEM and OSELM-BP, we show the numerical values of the various performance metrics in Table 8. Obviously, the best accomplished value was attained for OSELM-BP. After performing statistical differences, the overall t-test value was found to be less 0.05 which indicates statistical significance.

V. CONCLUSION

This article has presented a novel variant of extreme learning machine to solve the problem of random weights in the input and hidden layer. The variant uses the gradient of error as feed-back to correct the weights in the input hidden layer and in the hidden output layer for pre-defined number of iterations. Hence, it is designated as back-propagation online sequential extreme learning machine OSELM-BP. The variant was developed for IDS because it is one type of critical classification systems due to its security aspect. Hence, countering the random behaviour of input-hidden layer is crucial to prevent many types of false classifications.

The evaluation of the developed OSELM-BP was conducted on three datasets CICIDS-2017, KDD-99 and NSL-KDD. The configurations were based on changing the activation function and the number of hidden neurons of OSELM and the number of iterations of BP. Two set of results were used: the first one with using number of hidden neurons generated from characterization model and the second one based on lowest possible number of hidden neurons defined as two third of the number of features in the model. The finding is that OSELM-BP outperforms OSELM when the number of neurons is minimum, however, a degradation in the performance happens when the number of neurons is higher due to over-fitting. Future work of the article is to incorporate adaptive algorithm for selecting the appropriate number of hidden neurons to accomplish best possible performance and to enable automatic selection of number of iterations of OSELM-BP.

REFERENCES

[1] B. G. Atli, Y. Miche, A. Kalliola, I. Oliver, S. Holtmanns, and A. Lendasse, "Anomaly-based intrusion detection using extreme learning machine and aggregation of network traffic statistics in probability space," *Cognit. Comput.*, vol. 10, no. 5, pp. 848–863, Oct. 2018.

[2] S. Bosworth and M. E. Kabay, *Computer Security Handbook*. Hoboken, NJ, USA: Wiley, 2002.

[3] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2046–2069, 4th Quart., 2013.

[4] L. Wei, J. Zhang, F. Yuan, Y. Liu, J. Fan, and Q. Xu, "Vulnerability analysis for crypto devices against probing attack," in *Proc. 20th Asia South Pacific Design Automat. Conf.*, Jan. 2015, pp. 827–832.

[5] J. Andress, "Working with indicators of compromise," *ISSA J.*, pp. 14–20, 2015.

[6] L. A. Hughes and G. J. DeLone, "Viruses, worms, and trojan horses: Serious crimes, nuisance, or both?" *Social Sci. Comput. Rev.*, vol. 25, no. 1, pp. 78–98, Feb. 2007.

[7] C. Kruegel, F. Vigna, and G. Vigna, *Intrusion Detection and Correlation: Challenges and Solutions*. Cham, Switzerland: Springer, 2004.

[8] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *Int. J. Eng. Res. Technol.*, vol. 2, no. 12, pp. 1848–1853, 2013.

[9] J. Gao, L. Li, and B. Guo, "A new ExtendFace representation method for face recognition," *Neural Process. Lett.*, vol. 51, no. 1, pp. 473–486, Feb. 2020.

[10] X. Gao, Q. Sun, H. Xu, and J. Gao, "Sparse and collaborative representation based kernel pairwise linear regression for image set classification," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112886.

[11] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015.

[12] C. Zhang, J. Zhou, C. Li, W. Fu, and T. Peng, "A compound structure of ELM based on feature selection and parameter optimization using hybrid backtracking search algorithm for wind speed forecasting," *Energy Convers. Manage.*, vol. 143, pp. 360–376, Jul. 2017.

[13] J. Gao, L. Xu, A. Shi, and F. Huang, "A kernel-based block matrix decomposition approach for the classification of remotely sensed images," *Appl. Math. Comput.*, vol. 228, pp. 531–545, Feb. 2014.

[14] J. Gao and L. Xu, "An efficient method to solve the classification problem for remote sensing image," *AEU-Int. J. Electron. Commun.*, vol. 69, no. 1, pp. 198–205, Jan. 2015.

[15] J. Gao and L. Xu, "A novel spatial analysis method for remote sensing image classification," *Neural Process. Lett.*, vol. 43, no. 3, pp. 805–821, Jun. 2016.

[16] L. Li, H. Ge, and J. Gao, "A spectral-spatial kernel-based method for hyperspectral imagery classification," *Adv. Space Res.*, vol. 59, no. 4, pp. 954–967, Feb. 2017.

[17] C. Cheng, W. P. Tay, and G.-B. Huang, "Extreme learning machines for intrusion detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2012, pp. 1–8.

[18] R. Singh, H. Kumar, and R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8609–8624, Dec. 2015.

[19] K. J. R. A. W. S. L. Jebarani, "An efficient extreme learning machine based intrusion detection system," *GRD J., Global Res. Develop. J. Eng.*, 2016.

[20] Y. Li, R. Qiu, and S. Jing, "Intrusion detection system using online sequence extreme learning machine (OS-ELM) in advanced metering infrastructure of smart grid," *PLoS ONE*, vol. 13, no. 2, Feb. 2018, Art. no. e0192216.

[21] Y. Yu, S. Kang, and H. Qiu, "A new network intrusion detection algorithm: DA-ROS-ELM," *IEEJ Trans. Electr. Electron. Eng.*, vol. 13, no. 4, pp. 602–612, 2018.

[22] T. Matias, F. Souza, R. Araújo, N. Gonçalves, and J. P. Barreto, "On-line sequential extreme learning machine based on recursive partial least squares," *J. Process Control*, vol. 27, pp. 15–21, Mar. 2015.

[23] Z. Tian, G. Wang, Y. Ren, S. Li, and Y. Wang, "An adaptive online sequential extreme learning machine for short-term wind speed prediction based on improved artificial bee colony algorithm," *Neural Netw. World*, vol. 28, no. 3, pp. 191–212, 2018.

[24] Y. Li and Z. Yang, "Application of EOS-ELM with binary jaya-based feature selection to real-time transient stability assessment using PMU data," *IEEE Access*, vol. 5, pp. 23092–23101, 2017.

[25] S. Prabavathy, K. Sundarakantham, and S. M. Shalinie, "Design of cognitive fog computing for intrusion detection in Internet of Things," *J. Commun. Netw.*, vol. 20, no. 3, pp. 291–298, Jun. 2018.

[26] X. An, X. Zhou, X. Lü, F. Lin, and L. Yang, "Sample selected extreme learning machine based intrusion detection in fog computing and MEC," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–10, 2018.

[27] J. Gao, S. Chai, B. Zhang, and Y. Xia, "Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis," *Energies*, vol. 12, no. 7, p. 1223, Mar. 2019.

[28] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli, "A new intrusion detection system based on fast learning network and particle swarm optimization," *IEEE Access*, vol. 6, pp. 20255–20261, 2018.

[29] Y. Yu, Z. Ye, X. Zheng, and C. Rong, "An efficient cascaded method for network intrusion detection based on extreme learning machines," *J. Supercomput.*, vol. 74, no. 11, pp. 5797–5812, Nov. 2018.

[30] Y. Bazi, N. Alajlan, F. Melgani, H. AlHichri, S. Malek, and R. R. Yager, "Differential evolution extreme learning machine for the classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 6, pp. 1066–1070, Jun. 2014.

[31] D. R. Nayak, R. Dash, and B. Majhi, "An improved pathological brain detection system based on two-dimensional PCA and evolutionary extreme learning machine," *J. Med. Syst.*, vol. 42, no. 1, p. 19, Jan. 2018.

[32] P. Mohapatra, S. Chakravarty, and P. K. Dash, "An improved cuckoo search based extreme learning machine for medical data classification," *Swarm Evol. Comput.*, vol. 24, pp. 25–49, Oct. 2015.

[33] M. Akhavan-Amjadi, "Fetal electrocardiogram modeling using hybrid evolutionary firefly algorithm and extreme learning machine," *Multidimensional Syst. Signal Process.*, vol. 31, no. 1, pp. 117–133, Jan. 2020.

[34] T. Wu, M. Yao, and J. Yang, "Dolphin swarm extreme learning machine," *Cognit. Comput.*, vol. 9, no. 2, pp. 275–284, Apr. 2017.

[35] M. A. A. Albadr, S. Tiun, M. Ayob, and F. T. Al-Dhief, "Spoken language identification based on optimised genetic algorithm–extreme learning machine approach," *Int. J. Speech Technol.*, vol. 22, no. 3, pp. 711–727, Sep. 2019.

[36] M. A. A. Albadr, S. Tiun, F. T. Al-Dhief, and M. A. M. Sammour, "Spoken language identification based on the enhanced self-adjusting extreme learning machine approach," *PLoS ONE*, vol. 13, no. 4, Apr. 2018, Art. no. e0194770.

[37] G.-G. Wang, M. Lu, Y.-Q. Dong, and X.-J. Zhao, "Self-adaptive extreme learning machine," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 291–303, 2016.

[38] S. Lu, X. Qiu, J. Shi, N. Li, Z.-H. Lu, P. Chen, M.-M. Yang, F.-Y. Liu, W.-J. Jia, and Y. Zhang, "A pathological brain detection system based on extreme learning machine optimized by bat algorithm," *CNS Neurological Disorders-Drug Targets*, vol. 16, no. 1, pp. 23–29, Jan. 2017.

[39] Y. Lan, Y. C. Soh, and G.-B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, nos. 13–15, pp. 3391–3395, Aug. 2009.

[40] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving AdaBoost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset," *J. Phys., Conf. Ser.*, vol. 1192, Mar. 2019, Art. no. 012018.

[41] H. G. Kayacik, A. N. Zincir-Heywood, and M. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets," in *Proc. 3rd Annu. Conf. Privacy, Security Trust*, vol. 94, 2005, pp. 1722–1723.

[42] L. Dhanabal and S. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *Int. J. Adv. Res. Comput. Commun.*, vol. 4, no. 6, pp. 446–452, 2015.

[43] M. A. A. Albadr, S. Tiun, M. Ayob, and F. T. Al-Dhief, "Spoken language identification based on optimised genetic algorithm–extreme learning machine approach," *Int. J. Speech Technol.*, vol. 22, no. 3, pp. 711–727, Sep. 2019.

NEDHAL AHMAD HAMDI QAIWMCHI received the B.Sc. degree in software engineering foundation from Technical Education, Baghdad, Iraq, in 1998, and the M.S. degree from the Informatics Institute, University of Information Technology and Communications, Baghdad, in 2005. She is currently pursuing the Ph.D. degree with the Ferdowsi University of Mashhad, Iran. She was a Teacher with the Department of Software Engineering, Technical University of Baghdad, Iraq. Then, she transferred to the Technical Collage, Kirkuk, Iraq.

HALEH AMINTOOSI received the B.Sc. and M.Sc. degrees in computer engineering from the Ferdowsi University of Mashhad, Iran, in 2000 and 2003, respectively, and the Ph.D. degree in computer science from The University of New South Wales, Australia, in 2014. She is currently an Assistant Professor with the Department of Computer Engineering, Ferdowsi University of Mashhad. She is also a Visiting Senior Lecturer with the School of Computer Science and Engineering, The University of New South Wales. Her research interests include security, cryptography, and computer networks.

AMIRHOSSEIN MOHAJERZADEH received the B.S., M.S., and Ph.D. degrees from the Ferdowsi University of Mashhad, Mashhad, Iran, in 2005, 2007, and 2013, respectively. He has been a Computer Network Engineer with several networking projects with the Iran Telecommunication Research Center (ITRC), since 2008. He is currently an Assistant Professor with the Department of Computer Engineering, Ferdowsi University of Mashhad. He is the author of one book in Farsi language in networking field. He has published more than 30 international conference and journal papers. His research interests include cellular networks (5G), wireless sensor networks (WSNs), software-defined networking (SDN), smart grid, target tracking, modeling and analyzing computer networks, quality of service (QoS), and fuzzy logic control.