Appl Intell (2018) 48:2315–2327
https://doi.org/10.1007/s10489-017-1085-y

A new evolutionary neural networks based on intrusion detection systems using multiverse optimization

Ilyas Benmessahel[1] · Kun Xie[1] · Mouna Chellal[2]

Published online: 7 November 2017
© Springer Science+Business Media, LLC 2017

Abstract Building an intrusion detection system (IDS) has become an increasingly urgent issue for detecting network security breaches in computer and network systems. However, an effective and flexible IDS is imperative. In this paper, a new natural evolutionary algorithm (EA) called multiverse optimizer (MVO) is investigated and combined with an artificial neural network (ANN) to develop advanced detection approaches for an IDS. Under this context, the combination of ANN and EA produce evolutionary neural network (ENN). ENN makes this combination a new improved system for solving problems encountered by ANNs. The main idea of this work is to use an MVO to train a feed forward multilayer artificial neural network (MVO-ANN) to identify new attacks. This approach is applied to NSL-KDD and the new benchmark dataset called UNSW-NB15. In this manner, the effectiveness of our approach on detecting various forms of attack is demonstrated. Our results using UNSW-NB15 is better than those that were obtained using NSL-KDD. Furthermore, the efficacy of our proposed method is confirmed by performing better when compared to other well-known heuristic-based approaches such as practical swarm optimizer and artificial neural network (PSO-ANN).

Keywords Multilayer feed forward (MLFF) · Training neural network · Evolutionary algorithm (EA) · Multiverse optimization (MVO) · Intrusion detection systems (IDSs)

✉ Ilyas Benmessahel
ilyasbenms@hnu.edu.cn

Kun Xie
KunXie@hnu.edu.cn

Mouna Chellal
mounachellal@gmail.com

[1] College of Computer Science and Electronics Engineering, Hunan University, Changsha, China

[2] School of Information Science and Engineering, Central South University, Changsha, China

1 Introduction

The rapid development of the world wide web and the increase in local network systems have resulted in a continuous growth of internet users. Alongside these developments, a vast number of attack methods have been developed, and they have changed the computing world. This led to multitudes of potential vulnerabilities and cyber threats. A cyber-attack can be considered to constitute a modern warfare without weapons. It leads to disastrous and pernicious consequences, such as compromised systems, stolen and exposed personal information, unauthorized and illegal access, and data corruption. An effective permanent monitoring system, such as an intrusion detection system (IDS) is imperative in a computing network. The IDS ensures that the network is secure and guard against any security breach.

An IDS is a permanent monitoring system for computing networks. It is designed to monitor a targeted system by collecting audit data, analyze the gathered data, and establish a response plan before the environment is damaged. Two categories of intrusion detection techniques are commonly used, namely, anomaly-based and signature-based [3].

An anomaly-based IDS is based on audit trails. In an anomaly-based detection, the normal profile of the users is used, and any deviation is considered to be intrusive. The anomaly-based intrusion detection method has the capacity to detect a previously unknown attack. The signature-based IDS assumes that each attack has its own signature and can be identified by this signature [1]. Consequently, a signature-based IDS predetermines what is wrong and requires a knowledge base such as attack signature database. This type of IDS uses a pattern-matching method to identify an intrusion from the collected audit data, and then generate a response.

Many artificial intelligence (AI) techniques have been applied to signature-based and anomaly-based IDS categories. Rule-based approaches, machine learning, and data mining methods are direct approaches and efficient ways to implement these IDS. Data mining is the first proposed system for building an IDS [9]. Data mining is the process of extracting knowledge and useful information from an extensive database. It helps to extract rule patterns from a knowledge base and use them to predict future intrusion in similar datasets.

However, most of rule-based and data mining IDSs have limitations. They are unable to detect new attacks with new signatures since they don't have these signatures in their knowledge base. All new unknown attacks go unnoticed until the system has updated its knowledge base. However, the constant update of the rules in a knowledge base makes it difficult to manage and maintain this approach. The existing techniques are unable to detect the complex nature of the new attacks in networks. Machine learning approaches have been proposed in recent years to overcome these limitations [9].

Machine learning can handle problems that are encountered by rule-based approaches and data mining methods [11]. Machine learning is an AI field that includes advanced statistical methods for learning, prediction, and classification with multiple dependent and independent variables. The most commonly used machine learning approaches are support vector machine (SVM) [5], genetic algorithm (GA) [7], decision trees (DT) [17], and ANN [14].

ANNs present an attractive approach to intrusion detection that are capable of implementing a system that can learn from the data to understand the system users. This enables ANNs to deduce new information to provide a decision for generalizations. ANNs contain an important feature called learning by example which makes ANNs distinct from all conventional programming techniques (expert system). Unlike the traditional techniques, which are programmed, ANNs are trained. Until recently, many ANN-based IDS models have been developed, and each model has its own strengths.

This work focuses on handling the limitations of the ANNs. The ANN is one of the most widely used techniques and has successfully solved many complex practical problems that are difficult to solve through other methods. However, the general structure of ANNs still suffers from multiple problems [12, 14]. The ANN-based IDS has three major drawbacks.

– First, the error function of an ANN is a multimodal function that is frequently trapped into local minima.
– Second, this type of ANN-based IDS demonstrates a slow convergence.
– Third, over-fitting usually creates an overly complex model.

Note that the training process is a crucial procedure for learning the relationship between the inputs and the outputs and thus distinguishes this method from other types of conventional methods.

The well-known training algorithm, back-propagation (BP), is used to train the ANN's weights in a supervised mode. The back-propagation artificial neural network (BPANN) has been one of the most used algorithms for neural networks learning. However, one of the main challenges in the implementation of BPANN systems is the presence of many parameters (i.e., weight, activation function and gradient information) within the ANN structure. Specifically, the BPANN is likely to get into local structural minima, which negatively affects the capability of accurately assigning the ANN structures. This leaves unresolved problem of sub-optimal convergence to local minima.

To overcome the limitations attributed to the training algorithm BP and to avoid getting into local structural minima, we use the recently proposed MVO to train ANN. Hence, our proposed method can be classified as an ENN. The EA is the best solution for avoiding this limitation of the BPANN. Consequently, we can provide an effective and suitable alternative solution for the problem of ANN training algorithm and the multimodal search spaces. EAs constantly find a global solution optimum, whereas the BP algorithm can only find the initial point at the end of the slope of the search space (local optimum).

Moreover, we require a balanced dataset in which the distribution in the training set is not restricted to a certain class of attacks, which leads to overcoming the problem of over-fitting. In this proposed work, UNSW-NB15, the recently developed benchmark dataset for evaluating IDSs [15], is used to avoid the limitation of the above-mentioned dataset. UNSW-NB15 includes nine modern attacks and features of realistic normal traffic with a balanced distribution set.

In this work, we design a new ENN based on an IDS. We proposed the new MVO-ANN to improve the intrusion detection rate. This approach aims to solve the problem of the training algorithm of ANN and to evaluate the new proposed UNSW-NB15 dataset. This approach also solves the dataset-related problems. We applied our proposed method to enhance the detection of intrusions. The main contributions of this work can be summarized as follows:

- A new MVO is proposed to optimize the ANN model and to realize its effectiveness in handling the shortcomings of ANNs in the IDS field.
- The validity of the performance of the proposed model in detecting a new attack is clarified by using the new proposed UNSW-NB15 dataset and by comparing this dataset with the NSL-KDD dataset.
- A comparison with other EAs, namely, the PSO, is conducted using the UNSW-NB15 dataset to confirm the applicability of our model.
- We also compare our work with other techniques in the literature using the UNSW-NB15 dataset to validate the performance of our model.

Our method has the following advantages :

- Detects intrusions at a high accuracy rate.
- Potential to detect a new face of an attack.

The remainder of this paper is structured as follows: Section 2 reviews the related work. Section 3 describes the methodology and, then outlines the mathematical overview of the neural network and MVO. Section 4 discusses how the MVO can be deployed to train the ANN. Section 5 describes validation of the IDS. Section 6 presents the experimental setup and results. Section 7 summarizes the conclusions and provides directions for future research.

2 Related work

In this section, we briefly discuss the related work that is relevant to the ANNs used in IDSs, and then, we present related work for ENN.

The aforementioned IDSs can be divided into two typical categories, which are anomaly-based IDSs and signature-based IDSs. Signature-based IDSs use a knowledge base of network behavior. They can detect only a predetermined attack that is in the knowledge base of signature attacks, but they cannot detect unknown signature attacks. To accomplish that goal, they must update the knowledge base of signature attacks by adding the new signatures. Lee, Stolfo and Mok [9] were the first to apply data mining techniques in signature-based IDSs. This model comprises intuitive classification rules that can easily extract information from a dataset to detect intrusions.

Anomaly-based IDSs use a profile of the normal traffic activity. The anomaly-based approach can detect a new attack or any new potential attacks. This type of intrusion detection has difficulty in handling dynamic changes in the environment. The contemporary network traffic suffers from high false alarm rates because of the large boundaries between normal and abnormal behavior [27]. Zhang and Gu [10] designed an anomaly-based network IDS (NIDS) based on an artificial neural network multilayer perceptron (ANN-MLP) trained by a BP learning algorithm. Their proposed NIDS framework comprised three stages: the collection and preprocessing stage, training stage, and detection core. Their proposed system achieved a high detection rate.

Many researchers have investigated the deployment of ANNs for IDSs. Several ANN approaches are used in an enhanced IDS. In their work Ryan, Lin, and Miikkulainen [20], they developed an IDS based on the ANNMLP to identify a normal user profile. They asserted that each user leaves a print. A neural network, which is comparable to a detective, can be used to learn this print and identify each user.

Additionally, Sharawi, Sammany, El-Beltagy, and Saroit [21] developed an ANN-based IDS to resolve three classes of problems, namely, normal samples, attack samples, and the type of the attack. Their IDS exhibited high accuracy. The architecture of their ANN consists of two hidden layers, three outputs, and it was trained with a BP algorithm.

Moradi and Zulkernine [14] proposed an approach for ANNMLP-based IDS that can be used for intrusion detection on an offline mode. The results indicated that their proposed neural network model is highly accurate. The implemented ANNMLP consisted of a three-layer neural network with two hidden layers. A stopping criterion was used to stop the training process, thereby increasing the generalization capability of the neural network.

The ENN is a form of neural network in which evolution is fundamental in the optimization of its learning process. The EA is able to define a global searching capability and good approximate optimal solution. These properties make EA attractive to be applied in training neural network [4]. Several studies have been conducted and many systems were developed to detect intrusions through EAs.

In paper [22], Li used an EA to evolve neural network architectures and weights simultaneously to overcome the limitations of the existing BPANNs. The developed EA-based neural network achieved a higher detection rate than the BPANNs.

Michailidis, Katsikas, and Georgopoulos [12] presented an IDS using ENNs. The ENN is trained by a particle swarm optimization (PSO) algorithm to identify attacks and unknown attacks. The ENN trained by PSO showed an excellent classification rate.

In work [19], Rastegari proposed a novel architecture called evolving statistical rule sets for network intrusion detection (ESRNID). This architecture automatically generated rule sets based on the interval structure by analyzing and using as input the statistics of network traffic. ESRNID could address multiple types of attacks and outperform other existing methods.

Authors Wang, Yu, and Wang [25], proposed IDS based on ANN trained by GA. Their model, which was encoded in a binary system using a network audit dataset demonstrated a high detection rate.

Hofmann and Sick [8], applied an EA for the feature selection and evolutionary learning structure of a radial basis function network (RBFN). In their experiment, the network audit dataset had 137 typical examples as the number of packets, and the RBFN optimized using an EA achieved good results.

Additionally, Gonzalez, Gomez, Kaniganti, and Dasgupta [6] proposed an intrusion detection technique based on fuzzy rules generated with an EA. The condition part of the fuzzy detection rules was encoded with binary bits, and the fitness was evaluated using two factors, namely, the accuracy and the coverage of the rule. The performance of their intrusion detection technique was compared with those of other GA-based methods without fuzzy rules. Their technique was evaluated using two network audit datasets, their own wireless dataset and the KDD-Cup 99 dataset.

Han and Cho [7] proposed a novel intrusion detection technique based on ENNs. The model system was composed to BSM audit data and preprocessor monitors. Their GA modeler constructed normal behavior profiles, defined the genotype representation, identified appropriate genetic operators and evaluated the fitness. Their results proved that EAs are promising for training ANNs.

Consequently, based on the above mentioned work in the literature, we propose an IDS model built using the most promising MVO algorithm to train the ANN in solving the problems encountered by the ANNs training algorithm. Our proposed model is able to detect attacks and false alarm rates in UNSW-NB15 dataset with higher accuracy. This dataset contains a new emerging attack compared with the NSL-KDD dataset. Furthermore, our approach performs better than other techniques in the literature.

3 Methodology

This study designs and implements an IDS based on an ANN, that is trained by MVO, as shown in Fig. 1. Our objective is to design a novel framework that achieves the best global convergence and exhibits strong robustness using an EA for training the ANN.

ANNs can resolve several problems that are encountered by existing intrusion detection approaches. They identify the typical characteristics of the system users and identify statistically significant deviations from the established user behavior. In addition, an ANNs have open, flexible, and extendable structures. They can establish the general knowledge model of the behavior of an environment. The

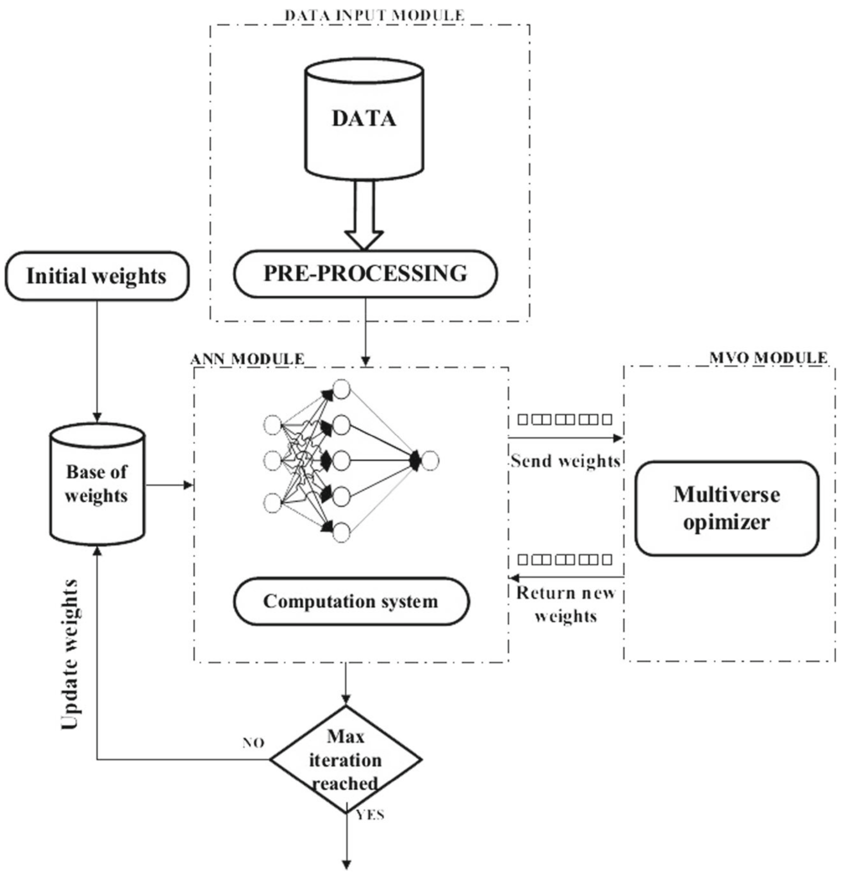


Fig. 1 Framework of the proposed IDS

procedure chiefly entails estimating the parameters of the neuron to enable the ANN to define the relation between the incoming pattern and the target output by training.

The training of a neural network is a complex task of great importance in problems that require training an ANN. It can be specified as an optimization problem. Finding a solution for the training procedure requires an answer derived from a linear constraint with a nonlinear optimization problem, and thus, different EAs are applied in the literature to solve this problem.

EAs are stochastic population search algorithms that aim to find an acceptable solution or approximately optimal solutions in multimodal search spaces. EAs have strong global convergence and strong robustness. They can address the problem of being trapped in a local minimum in the multimodal search space before reaching the global optimum with fast convergence. Figure 1 illustrates the framework of the proposed IDS and shows that it can be divided into three principal modules, namely, the data input, ANN network and the MVO module, which can be described as follows:

For the first step, in our framework, the data input module is used. This module is responsible for processing, filtering, and extracting the features from the audit data. The dataset contains predefined training and testing sets that are used as inputs for the next ANN module. Before the data are fed into the ANN module, the data input module should be mapped in the incoming inputs to fall between zero and one [0, 1], to render these data usable for the next module.

In the second step, the ANN module receives (N) training attributes of the input data dimension from the data input module. The ANN module is designed as an MLP, which is a feed-forward neural network with the architecture of one input layer, one hidden layer, and one output layer. The incoming inputs from the data input module (the training dataset) are fed into the ANN module as an input training pattern for training the ANN. This training process is performed by sending the weights to the MVO module.

The MVO module is designed as a standalone system for updating the synaptic weights after each iteration. In each iteration of the training process, the MVO module sends its individuals as a set of weights into an ANN module, which evaluates these individuals based on a training dataset and then returns their fitness values. In this work, the mean squared error (MSE) is selected as a known fitness function for the proposed MVO training algorithm. The synaptic weights are obtained by minimizing the value of MSE. The training process stops when the maximum number of iterations is reached. Afterward, the knowledge base (weights and biases) is updated.

In the third step, once the ANN is trained with the training dataset, the testing inputs are fed from the testing dataset into the trained ANN to predict the output. The testing process of the ANN can be viewed as checking the predicted output with the closest match to any of the target classes.

3.1 Artificial neural network module

To optimize the generalization capability of the ANN module, we selected an ANNMLP with one hidden layer and binary classification because of its structure.

The ANN comprises of parallel processing elements, which are highly interconnected, and distributes a set of

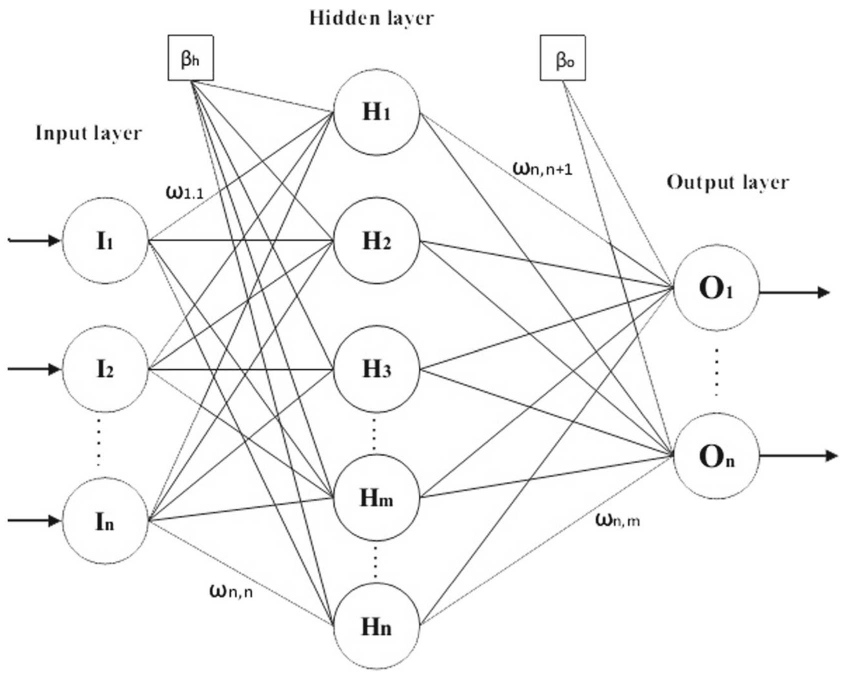


Fig. 2 Simple architecture of the ANN

inputs to a set of desired outputs. In the ANN, the neurons are arranged in layers through unidirectional branches in the forward direction. The ANN model compromises of three layers: the input layer, fixed hidden layers and the output layer [28]. An example of a simple ANN with a single hidden layer is shown in Fig. 2.

The input layer has a number of neurons that correspond to the number of input attributes, but the number of hidden layers is limited to a single hidden layer with M biases. The hidden neurons are connected to the input neurons at the initial weights. During the training process, these weights are adjusted while the neural network model is constructed. The output layer is the third layer, which includes only one neuron for each class. The output neurons can be assigned according to the expected classification result values; 1 for a normal connection (correct class) and 0 for an abnormal connection (incorrect class) with a single bias.

Usually, each input fed to the network is multiplied by the corresponding weight, and the sum of the products produce a weighted sum function; next the resulting sum is passed through a transfer function called the activation function. The sum function is calculated by adding the products of the coming inputs, the initial weights and the added bias weight, as shown in (1). In (1), ωij is the connection weight linked to the input neuron Ii to neuron j, βj is the added bias weight, and n is the total number of neurons in the input layer.

$$S_j = \sum_{i=1}^{K} \omega_{ij} * I + \beta_j \tag{1}$$

3.2 Multiverse optimization module

To optimize the exploration, exploitation and approximation search capacity of the EA, we used a multiverse optimizer in the MVO module. The MVO was proposed by Seyedali, Seyed, and Abdolreza [13].

The MVO algorithm is based on three cosmological concepts: white holes, which are not actually seen in the universe; black holes, which are the opposite of white holes, and wormholes, which are interconnected parts of the universe. These three concepts are the main inspiration of the MVO algorithm, which simulates the movement and interaction between universes through black holes, white holes, and wormholes. The MVO algorithm divides the local search process into the exploration process and exploitation process [4].

In MVO, a solution refers to as a final best universe, a variable in the solution corresponds to an object in the universe, the inflation rate of a solution corresponds to the fitness of the solution and the term time corresponds to the iteration.

The following rules are applied to all the universes during the optimization process:

– When the inflation rate is higher, the probability of an existing white hole is higher.
– When the inflation rate is lower, the probability of an existing black hole is lower.
– The best universe is formed from random movements of objects via wormholes.

In the exchange of objects between universes, a universe with a higher inflation rate sends objects to other universes that have a lower inflation rate. Furthermore, a universe with a lower inflation rate accepts more objects from better universes to reach a stable status and becomes the best universe with an improved inflation rate. This exchange process can assure the improvement of the average inflation rates of all of the universes during the time conducted [2].

During the optimization, the universes are sorted according to their inflation rates, and one is selected, using the roulette wheel, to be white hole. The initiation of the MVO can be described according to the following expression:

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^d \end{bmatrix}$$

$$x_i^j = \begin{cases} x_k^j & r1 < NI(Ui) \\ x_k^j & r1 \geq NI(Ui) \end{cases}$$

where U the matrixes are the set of universes, n is the number of universes, d is number of optimized parameters, X_i^j is the j^{th} parameter of the ith universe, X_k^j is the j^{th} parameter of the k^{th} universe selected by a roulette wheel; UI is the ith universe, $r1$ is a random number in the interval of [0,1], and $NI(UI)$ is the normalized inflation rate of the i^{th} universe.

To maintain an exchange between the local universe and improving the inflation rate, suppose that wormhole tunnels can deliver random exchange of objects among all of the universes toward the best universe formed thus far. The formula for modeling this mechanism is the following:

$$X_i^j = \begin{cases} \begin{cases} x_j + TDR + ((ub_j - lb_j) * r4 + lb_j) & (r3) < WEP \\ x_j + TDR + ((ub_j - lb_j) * r4 + lb_j) & (r3) \geq WEP \end{cases} & (r2) < WEP \\ X_i^j & (r2) \geq WEP \end{cases} \tag{2}$$

where x_j indicates the j^{th} in the formed universe; x_i^j represents the j^{th} variable in the i^{th} universe; the traveling distance rate (TDR) and wormhole existence probability (WEP) are the two main coefficients; lb_j represents the lowest j^{th} variable, ub_j denotes the highest j^{th} variable, and the values of $r2, r3, r4$ are random variables in the interval [0, 1]. The formulas for the two coefficients are given by

$$TDR = 1 - \left(\frac{l^{\frac{1}{p}}}{L^{\frac{1}{p}}} \right) \tag{3}$$

$$WEP = min - l * \left(\frac{min - max}{L} \right) \tag{4}$$

where p (=6) denotes the accuracy of exploitation over the iterations. where l indicates the current iteration, and L shows the maximum number of iterations.

WEP and TDR are increased at every iteration to achieve a more precise exploitation/local search around the best obtained universe. High accuracy exploration and exploitation processes are the benefits of this algorithm [4], and these benefits motivated us to use a novel training algorithm based on this MVO mechanism. The general steps of the MVO algorithm are presented in Algorithm 1:

Algorithm 1 Multiverse Optimization Algorithm

```
1: Initialize the parameters of the MVO: lb, ub.
2: Create a set of random universes based on ub and lb.
3: Calculate the corresponding inflation rate (fitness) for each
   universe.
4: Calculate WEP, TDR by equations 3, 4.
5: Exchange objects between universes (higher to lower inflation
   rate).
6: Objects in each universe teleport to the best universe using
   equation 2.
7: Go to step 3 if the end criterion is not satisfied.
8: Return the best universe formed thus far.
```

The optimization process begins by creating and initializing the parameters, such as defining the size of the population and the upper and lower bounds. Then we initialize a set of random universes based on the upper and lower bounds. The corresponding inflation rate (fitness value) is calculated for each universe to define the best possible inflation rate. Afterward, at each iteration, objects in the universes with high inflation rates are inclined to shift to the universes with low inflation rates through white or black holes. Simultaneously, objects in each universe randomly move to the best universe through wormholes. Finally, the best universe unit is created at the end of the operation.

4 Training of ANN with the MVO algorithm

The training process is an important phase for the optimization of the ANN. The objectives of this process is to search for the synaptic weights of the ANN and to reduce the MSE, which represents the cost function of the ANN.

The MVO algorithm starts the optimization process by generating a population of solutions and assumes that each universe is an individual in the solution population. The population is generated randomly. The size of the solution represents the problem's dimension. The representation and design of the MVO individuals are important factors in training the ANN.

In the ANN training, each individual represents all of the weights and biases of the ANN structure. The objective of the training process is to find the proper values for the weights and biases and consequently minimize ANN error and achieve the highest classification and prediction accuracy. In our implementation, each individual in MVO is a vector that include the connection weights between ANN layers. The MVO individual is encoded as below:

$$\overrightarrow{V} = \{\overrightarrow{\omega}, \overrightarrow{\beta}\} = \{\omega_{1,1}, \omega_{1,2} \cdots, \omega_{n,m}, \beta_1, \beta_2 \cdots, \beta_n\} \tag{5}$$

where n is the total number of input neurons, ω_{ij} is the linked weight from the i^{th} neuron to the j^{th} neuron, and θ_j is the added bias of the hidden layer neuron as illustrated in Fig. 3.

The number of objects in each individual is calculated as follows:

$$Indv_{nbr} = (n * m) + (2 * m) + 1 \tag{6}$$

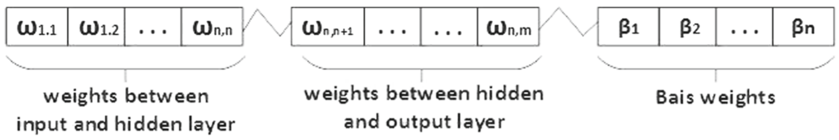
In this paper, MSE is used as the principal cost function of the proposed MVO training algorithm. The training goal is to minimize the MSE until the maximum number of generations has been reached. The MSE can be calculated by:

$$MSE = (1/T_n) * \sum_{i=1}^{T_n} output - input \tag{7}$$

where input is the actual data and output is the estimated values and T_n is the number of instances in the dataset.

In Fig. 4, the individual vector is represented by three parts. The first part is the set of weights that link the input and hidden layer. The second part is a set of weights that link the hidden and output layer. The third part is the set of biases.

The general training pseudo-code of the MVO-ANN is presented in Algorithm 2. In lines 1 and 2, the general algorithm parameters, namely, MaxIteration, Nb of universe,



$\omega_{1.1}$ $\omega_{1.2}$ ... $\omega_{n,n}$
$\omega_{n,n+1}$ ... ... $\omega_{n,m}$
β_1 β_2 ... β_n
weights between input and hidden layer
weights between hidden and output layer
Bais weights

Fig. 3 MVO individual representation

and the upper and lower bounds, are inputted, a set of individuals is generated randomly, where the dimension of the individual is defined by equation 6. In lines 3-8, the MSE for each individual in the whole universe is checked and the MSE of the global minimal is derived. In line 9, all the loops are assigned to Max-iteration. In line 10, the MVO parameters, namely, TDR and WEP, are calculated by equations 4, 3, respectively. In lines 11-15, the optimization process is started and the objects in the individual universe with high inflation rates incline to shift to the universe with low inflation rates through white or black holes. Simultaneously, objects in each universe randomly move to the best universe via wormholes by equation 3. In lines 16-21, we calculate the new MSE for a set of individuals, and the MSE of the global minimum is derived. In line 22, we save the current best individual, and in line 24, the best individual (solution) is identified.

Algorithm 2 MVO Training ANN Pseudo-Code

```
1: Initialize the training parameters: WEP, TDR, lb, ub, Max-iteration, Nbr-of-individuals.
2: Create a set of random individuals based on the problem dimension via equation 6.
3: for each individual do
4:     Calculate the MSE for the individual by equation 7
5:     if the current MSE $<$ the global minimal MSE then
6:         Update the global minimal MSE.
7:     end if
8: end for
9: for iteration (t) $<=$ Max-iteration do do
10:     for each individual do
11:         Calculate the parameter of MVO: WEP, TDR.
12:         Run the optimization process
13:         Exchange objects between the universes (higher to lower inflation rate).
14:         Objects in each universe teleport to the best universe using equation 2
15:     end for
16:     for each individual do
17:         Calculate the MSE for individual by equation 7
18:         if the new current MSE $<$ global minimal MSE then
19:             Update the globally minimal MSE
20:         end if
21:     end for
22:     Save the current best solution with the minimal MSE
23: end forreturn The best solution of the minimal MSE.
```

5 Validation of the intrusion detection system

The effectiveness of the use of the MVOANN approach to the IDS was evaluated, and its performance was compared with those of other existing IDS techniques. For the validation, publicly available datasets, namely, KDD Cup 99, NSL-KDD, and the new dataset UNSW-NB15, were used. These datasets are dedicated for the offline evaluation of the IDSs.

5.1 UNSW-NB15 dataset

The UNSW-NB15 dataset has been recently released [24]. This dataset contains nine different modern attack types and a wide variety of real normal activities [16].

The dataset contains real modern normal behaviors and contemporary synthesized attack activities and consists of 49 features with their class labels. This dataset comprises 2,540,044 observations. In this study, the UNSW-NB15 was divided into a training set and testing set. Furthermore, this new dataset ensures an accurate evaluation of IDSs [15]. Table 1 shows the distribution of the records in the UNSW-NB15 dataset.

5.2 KDD dataset

The KDD Cup 99, which is a processed version of the DARPA 98 dataset, has been the most widely used benchmark dataset for IDS evaluation.

The total number of connection records in the training dataset is approximately 5 million. However, this dataset has a number of redundant instances in the training and testing sets. These redundancies significantly affect the performance and cause a poor evaluation of the IDSs. Thus, the NSL-KDD dataset was constructed to enhance the KDD Cup 99 dataset and solve the inherent problems of the latter [23].

This dataset is formed from the different parts of the original KDD Cup 99 dataset, without the redundancies and duplications. In addition, the problem of having an unbalanced distribution in each class, either in the training set or the testing set, was solved, to improve the accuracy of the IDS evaluation. The NSL-KDD dataset includes 41 attributes, which are labeled normal connections or attack types. The NSL-KDD dataset is divided into training and testing sets, and it has four attack classes: DoS, U2R, R2L, and probe [18]. Table 2 shows the distribution of the records in the NSL-KDD dataset for the training and testing sets.

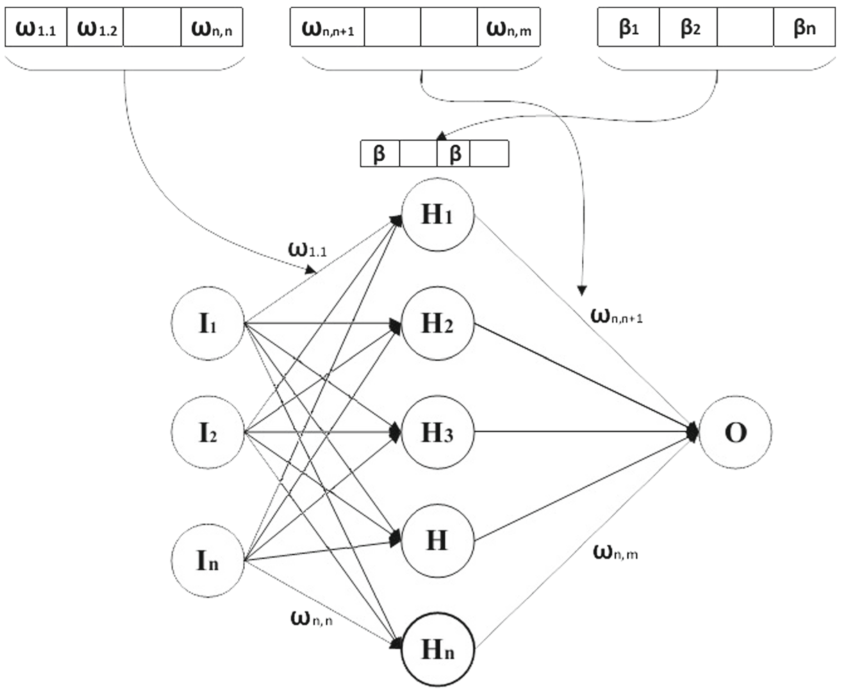


Fig. 4 Mapping an MVO individual representation to an ANN network

6 Experimental setup and results

This section presents a comprehensive evaluation of the MVO-ANN. The performance of the MVO trained ANN is evaluated on the NSL-KDD dataset, the UNSW-NB15 dataset and compared with existing techniques.

6.1 Experimental setup

The proposed model was implemented and evaluated in Visual Basic 2010 on a personal PC with Core I5 2.4 GHz CPU and 4 GB RAM.

The common parameter settings were used for the MVO. The appropriate setting of parameters significantly influences the performance evaluation of the proposed model. In the MVO training using the UNWS-NB15 dataset, the following parameters were the following: minimum WEP=0.2; maximum WEP=1; number of individuals=5; and number of iterations=100. When the NSL-KDD dataset was used, the parameters were the following: minimum WEP=0.2; maximum WEP=0.6; number of individuals=4; and number of iterations=100.

In the experiment, all of the feature inputs were mapped using the same scale [0, 1]. Min-max normalization was applied to the two datasets given by (8), where X is the normalized value of x in the range of [0, 1].

$$X_i^1 = \frac{x_i - min_v}{max_v - min_v} \tag{8}$$

The IDS was evaluated on several factors. The main factors include the detection rate (DR), the false alarm rate (FAR), and the accuracy (ACC). The FAR, DR, and ACC are calculated based on the main instances: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). They are included in the confusion matrix (CM). The CM has a dimension of NN, and it presents the classification results. Table 3 illustrates a 2∗2 CM.

The abbreviations of the confusion matrix are as follows:

TP: Number of connections successfully classified as anomalies by the classifier.

Table 1 Statistics of the UNSW-NB15 dataset

	Train UNSW-NB15		Test UNSW-NB15	
	Effective	%	Effective	%
Normal	65000	37.08	37000	44.94
Attack	110341	62.92	45332	55.06
Total	175341	100	82332	100

Table 2 Statistics of the NSL-KDD data set

	Learning NSLKDD				Test NSLKDD	
	20% NSLKDD		Total NSLKDD			
	Effective	%	Effective	%	Effective	%
Normal	13449	53.38	67244	53.38	9714	44.5
Attack	11743	62.92	58730	46.62	12830	55.4
Total	25192	100	125974	100	22544	1000

Table 3 Confusion matrix

		Predicted		Total
		Normal	Attacks	
Actual	Normal	TN	FP	TN+FN
	Attacks	FN	TP	TP+FN
Total		TN+FN	TP+FP	

FN: Number of anomalous connections that are misclassified as normal by the classifier.

FP: Number of normal/non-intrusive connections that are misclassified as intrusive as anomalies by the classifier.

TN: Number of normal/non-intrusive connections that are successfully classified as normal by the classifier.

Different performance metrics can be derived from the CM by using the CM variables, such as the precision. However, the most common performance metrics are the detection accuracy (DA), true positive rate (TPR), and false positive rate (FPR). The descriptions of these metrics are as follows:

- TPR: TPR is also known as DR or recall. It is the ratio of the number of successfully classified anomalies to the total number of connections. It indicates the precision of the model in detecting anomalies from all of the anomalous connections. It can be derived as follows:

$$TPR = \frac{TP}{TP + FN} \tag{9}$$

- FPR: FPR is also known as the false alarm rate (FAR). It can be computed by dividing the number of normal connections that are misclassified as anomalies by the total number of normal connections. It can be derived as follows:

$$FPR = \frac{FP}{FP + TN} \tag{10}$$

- DA: DA is the percentage of correctly classified connections to the total number of connections. It can be derived as follows:

$$DA = \frac{TP + TN}{TP + FN + FP + TN} \tag{11}$$

Table 4 Distribution statistics of 20% of the NSL-KDD training dataset

Normal	DoS	Prob	R2L	U2R
13449	9234	2289	209	11

Table 5 Distribution statistics of 80% of the NSL-KDD testing dataset

Normal	DoS	Prob	R2L	U2R
9443	6421	1912	209	67

6.2 Results

The performance of the proposed MVO-ANN based IDS was evaluated through a series of experiments.

The NSL-KDD dataset was used in the first phase of the experiment. Only 20% of the NSL-KDD training dataset was used. This dataset contains the signatures of different types (normal connections and four attack types, i.e., DOS, R2L, U2R, and PROBE). The NSL-KDD training dataset was used to train the proposed MVO-ANN model. The statistics of the NSL-KDD training dataset, which presents only 20% of the global NSL-KDD dataset, is shown in Table 4.

For the test data, only 80% of the NSL-KDD testing dataset was used. The statistics of this dataset are in Table 5.

After training the MVO-ANN, the evaluation using the NSL-KDD testing dataset was conducted. Table 6 shows the performance test results in the first experimental phase. Figure 5 illustrates the performance of the proposed method during the training process with the NSL-KDD training dataset; MSE was used as the performance evaluation factor.

In the second phase of the experiment, we used the UNSW-NB15 dataset to evaluate the performance of the proposed method. This dataset contains signatures of different types (normal connections and nine attack types, i.e., fuzzers, analysis, backdoor, DoS, exploit, generic, reconnaissance, shellcode, and worm). The training dataset was used to train the proposed method. The statistical distribution of the UNSW-NB15 training dataset is shown in Table 7.

The UNSW-NB15 testing dataset was used for the evaluation. The statistics of this dataset are shown in Table 8.

After training the MVO-ANN model, the evaluation using the UNSW-NB15 testing dataset was launched. Table 9

Table 6 Performance test results of MVO-ANN for NSL-KDD

ACC		DR		FAR	
Rate	Effective	Rate	Effective	Rate	Effective
98.21%	17730	96.25%	8275	0.032%	303

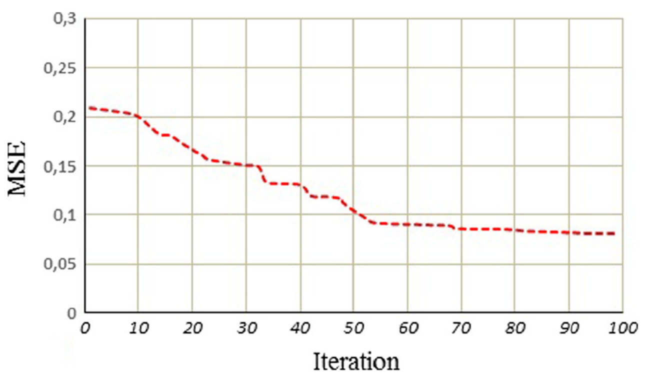


Fig. 5 MSE convergence curve of the MVO-ANN for NSL-KDD

shows the performance test results in the second experimental phase. Figure 6 illustrates the performance of the proposed method during the training process with the UNSW-NB15 training dataset, MSE was used as the performance evaluation factor.

Instead of presenting DA only in our results, we also included TPR and FPR. Table 6 shows the performance test results of the MVO-ANN for the NSL-KDD database. The model obtained the highest DA (98.21%), highest TPR (96.25%), and lowest FPR (0.032%).

For the UNSW-NB15 dataset, Table 9 shows that MVO-ANN likewise obtained the highest DA (99.61%), the highest TPR (99.65%), and the lowest FPR (0.004%).

The results indicate that the proposed model trained with the UNSW-NB15 training dataset achieves better performance. Figures 6 and 5 show the evolution curve of the MSE function of our model for the NSL-KDD dataset and the UNSW-NB15 dataset, respectively. Figure 7 shows the evolution of the converging curves of the MSEs within 100 iterations.

The results indicate that the ANN trained using the UNSW-NB15 dataset exhibits enhanced capability in avoiding being trapped into local minima. In addition, the convergence speed is faster when the UNSW-NB15 dataset is used than when the NSL-KDD dataset is used.

Overall, the results indicate that the evaluation curve for the UNSW-NB15 dataset is better than that for the NSL-KDD dataset. The UNSW-NB15 dataset is considered to be contemporary with behaviors similar to a modern attack and behaviors similar to a normal network traffic as well as dissimilarity between a modern attack and normal network behavior. As a consequence, the UNSW-NB15 dataset can reliably test the existing methods for IDS.

Furthermore, UNSW-NB15 has a wide variety of values that represent the nature of a modern real network, and these values are similar between the attack and normal records. The data distribution of the UNSW-NB15 dataset between the training set and testing set is balanced because all of the observations were generated from the same test-bed. As a result, the search capability and exploitation performance are enhanced. In contrast, the data distribution of the NSL-KDD dataset between the training set and testing set is unbalanced because of the addition of new attacks to the testing set. Consequently, the model failed to distinguish normal behavior from attack observations and exhibited partially reduced exploration and exploitation capabilities.

Based on the results achieved in the first and the second phase of this experimentation, we can determine that the UNSW-NB15 dataset has better performance than the NSL-KDD dataset. We test the applicability of the proposed methods using UNSW-NB15 when compared with PSO-ANN using UNSW-NB15 dataset.

In the third phase we used the UNSW-NB15 dataset to compare the performance of the MVO-ANN and PSO-ANN.

The performance test results in the third experimental phase are shown in Table 10. Figure 8 illustrates the evolution of the converging curves of the MSEs within 100 iterations.

The algorithms for training ANNs do not need only strong exploration ability, but also precise exploitation ability. The results of the classification accuracy, DA and FPR obtained by PSO-ANN and MVO-ANN, it is shown that MVO-ANN performs better than PSO-ANN due to the more

Table 7 Distribution statistics of the UNSW-NB15 training dataset

Normal	Fuzzers	Analysis	Backdoor	DoS
56000	18184	2000	1746	12264
Exploit	Generic	Reconnaissance	Shellcode	Worm
33393	40000	10491	1133	130

Table 8 Distribution statistics of the UNSW-NB15 testing dataset

Normal	Fuzzers	Analysis	Backdoor	DoS
37000	6062	677	583	4089
Exploit	Generic	Reconnaissance	Shellcode	Worm
11132	18871	3496	378	44

Table 9 Performance test results of the MVO-ANN for UNSW-NB15

AC		DR		FAR	
Rate	Effective	Rate	Effective	Rate	Effective
99.61%	82008	99.65%	45173	0.004%	159

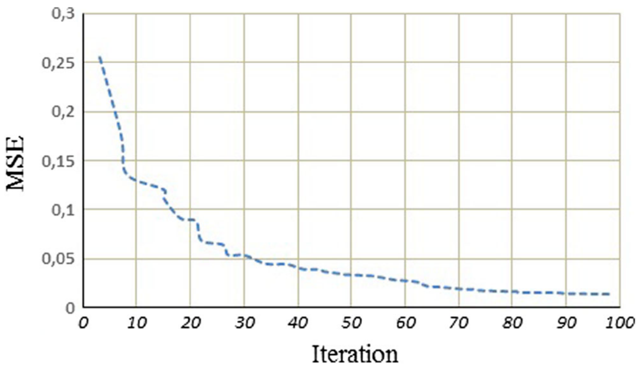


Fig. 6 MSE convergence curve of MVO-ANN for UNSW-NB15

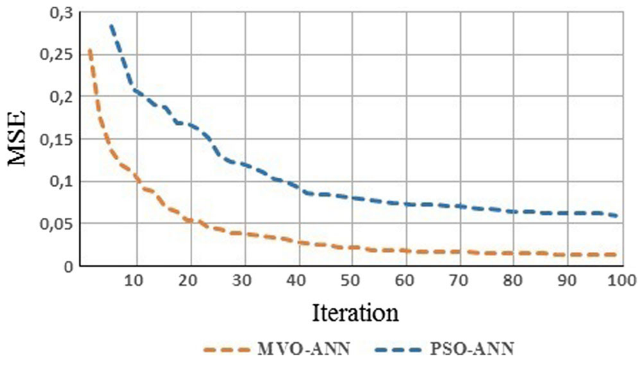


Fig. 8 MSE convergence curves of PSO-ANN and MVO-ANN for UNSW-NB15

precise exploitation ability of MVO. PSO-ANN still suffers from the problem of becoming trapped in local minima. This weakness means that PSO-ANN has unstable performance. The results obtained by MVO-ANN prove that it has both strong exploitation and good exploration abilities. In other words, the strength of PSO and PSO has been successfully utilized and gives outstanding performance in the ANN training. This result means that MVO-ANN is capable of solving the problem of becoming trapped in local minima and gives a fast convergence speed.

The comparisons of the performance results of the proposed model and other models for the UNSW-NB15 dataset and NSL-KDD dataset are shown in Table 11 and 12, respectively. The proposed model clearly performs the best in terms of DA and FPR. The data correctly classified by the proposed model are more than those correctly classified by the static models. Furthermore, MVO-ANN exhibits a significantly lower FPR.

Therefore, the proposed method offers the best strong exploration and precise exploitation capabilities. The proposed training algorithm MVO is effective and feasible for application to IDS research.

Table 10 Performance test results of PSO-ANN for UNSW-NB15

ACC		DR		FAR	
Rate	Effective	Rate	Effective	Rate	Effective
91.87%	75640	98.61%	44486	0.0186%	846

Table 11 Comparison of the proposed model with other models using UNSW-NB15 in terms of the DA and FPR

Techniques	Previous Work	Accuracy (%)	FAR (%)
ANN(MLP)	[17, 26]	81.34	21.13
LR(Logistic Regression)		83.15	18.84
DT(Decision Tree)		85.56	15.78
NB(Naïve Bayes)		82.07	18.56
EM(Expectation Max)		78.47	23.79
ANN(MVO-ANN)	in this work	99.61	0.004

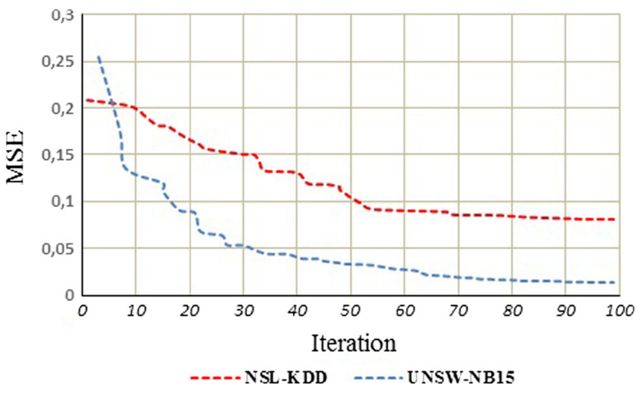


Fig. 7 MSE convergence curves of the MVO-ANN for UNSW-NB15 and NSL-KDD

Table 12 Comparison of the proposed model with other models using NSL-KDD in terms of the DA and FPR

Techniques	Previous Work	Accuracy (%)	FAR (%)
ANN(MLP)	[17, 26]	97.04	1.48
LR(Logistic Regression)		92.75	18.84
DT(Decision Tree)		–	–
NB(Naïve Bayes)		95	5
EM(Expectation Max)		–	–
ANN(MVO-ANN)	in this work	98.21	0.032

7 Conclusion

This paper presents a new ENN, namely, an MVO trained ANN. It focuses on the applicability of the new cosmology-inspired algorithm MVO to train ANNs. The MSE and DA of the proposed model have been obtained using the NSL-KDD and UNSW-NB15 datasets. The performance of the model was compared with those of popular intrusion detection techniques, based on such as PSO-ANN, ANN-MLP and classification techniques. The MVO-ANN trained with the UNSW-NB15 dataset and that trained with the NSL-KDD dataset obtained DAs of 99.61% and 98.21%, respectively. These values are higher than those obtained by other methods tested using the UNSW-NB15 and NSL-KDD datasets. The results show the potential applicability of ENN for developing practical IDSs. However, this study has mainly evaluated the models according to the feature intrusion detection datasets; an adequate feature selection technique has not been selected. Therefore, our future work will focus on minimizing the number of selected features and applying other ENN models to develop an effective IDS.

References

1. Ahmed M, Mahmood AN, Hu J (2016) A survey of network anomaly detection techniques. J Netw Comput Appl 60(C): 19–31
2. Ali EE, El-Hameed MA, El-Fergany AA, El-Arini MM (2016) Parameter extraction of photovoltaic generating units using multi-verse optimizer. Sustainable Energy Technol Assess 17:68–76
3. Bamakan SMH, Amiri B, Mirzabagheri M, Shi Y (2015) A new intrusion detection approach using pso based multiple criteria linear programming. Procedia Comput Sci 55:231–237
4. Ding S, Li H, Su C, Yu J, Jin F (2013) Evolutionary artificial neural networks: a review. Artif Intell Rev 39(3):251–260
5. Enache AC, Patriciu VV (2014) Intrusions detection based on support vector machine optimized with swarm intelligence. In: 2014 IEEE 9th IEEE international symposium on applied computational intelligence and informatics (SACI), pp 153–158
6. Gonzalez F, Gomez J, Kaniganti M, Dasgupta D (2003) An evolutionary approach to generate fuzzy anomaly (attack) signatures. In: IEEE systems man and cybernetics society information assurance workshop, 2003, pp 251–259
7. Han S-J, Cho S-B (2005) Evolutionary neural networks for anomaly detection based on the behavior of a program. IEEE Trans Syst Man Cybern B Cybern 36(3):559–570
8. Hofmann A, Sick B (2003) Evolutionary optimization of radial basis function networks for intrusion detection. In: Proceedings of the international joint conference on neural networks, 2003., vol 1, vol 1, pp 415–420
9. Lee W, Stolfo SJ, Mok KW (2000) Adaptive intrusion detection: A data mining approach. Artif Intell Rev 14(6):533–567
10. Li J, Zhang G-Y, Gu G-C (2004) The research and implementation of intelligent intrusion detection system based on artificial neural network. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826), vol 5, pp 3178–3182
11. Lin W-C, Ke S-W, Tsai C-F (2015) Cann: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowl-Based Syst 78:13–21
12. Michailidis E, Katsikas SK, Georgopoulos E (2008) Intrusion detection using evolutionary neural networks. In: 2008 Panhellenic Conference on Informatics, pp 8–12
13. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Applic 27(2):495–513
14. Moradi M, Zulkernine M (2004) A neural network based system for intrusion detection and classification of attacks. In: Proceedings of the IEEE international conference on advances in intelligent systems, pp 15–18
15. Moustafa N, Slay J (2015) Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 military communications and information systems conference (MilCIS), pp 1–6
16. Moustafa N, Slay J (2015) The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems. In: 2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS), pp 25–31. IEEE
17. Moustafa N, Slay J (2016) The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. Inf Secur J A Global Perspect 25(1-3):18–31
18. NSLKDD (2013) Nsl-kdd dataset, retrieved from http://nsl.cs.unb.ca/NSL-KDD
19. Rastegari S (2015) Intelligent network intrusion detection using an evolutionary computation approach
20. Ryan J, Lin M-J, Miikkulainen R (1998) Intrusion detection with neural networks. In: Jordan MI, Kearns MJ, Solla SA (eds) Advances in Neural Information Processing Systems 10, pages 943–949. MIT Press
21. Sammany M, Sharawi M, El-Beltagy M, Saroit I (2007) Artificial neural networks architecture for intrusion detection systems and classification of attacks. In: The 5th international conference INFO2007, pp 24–26
22. Shuhui LI (2010) Improved evolutionary neural network algorithm and its applications in intrusion detection. Mod Electron Technique 1:028
23. Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the kdd cup 99 data set. In: Proceedings of the 2nd IEEE international conference on computational intelligence for security and defense applications, CISDA'09, pp 53–58, Piscataway, NJ, USA. IEEE Press
24. the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) (2015) Unsw-nb15 dataset, retrieved from http://www.accs.unsw.adfa.edu.au/
25. Wang L, Yu G, Wang G, Wang D (2001) Method of evolutionary neural network-based intrusion detection. In: 2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No.01EX479), vol 5, pp 13–18
26. Witten IH, Frank E, Hall MA (2011). In: Witten IH, Frank E, Hall MA (eds) Data Mining: Practical Machine Learning Tools and Techniques (3rd Edition), The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, Boston, third edition edition
27. Wu SX, Banzhaf W (2010) The use of computational intelligence in intrusion detection systems: A review. Appl Soft Comput 10(1):1–35