IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 27, 2020, accepted March 17, 2020, date of publication April 7, 2020, date of current version April 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986217

An Efficient Anomaly Intrusion Detection Method With Feature Selection and Evolutionary Neural Network

SAMIRA SARVARI[1], NOR FAZLIDA MOHD SANI[1], ZURINA MOHD HANAPI[2], AND MOHD TAUFIK ABDULLAH[1]

[1]Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Malaysia
[2]Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang 43400, Malaysia

Corresponding author: Nor Fazlida Mohd Sani (fazlida@upm.edu.my)

This work was supported by the Universiti Putra Malaysia (UPM).

ABSTRACT Recently, with the technological and digital revolution, the security of data is very crucial as a massive amount of data is generated from various networks. Intrusion Detection System (IDS) has been observed to be perhaps the best solution because of its capability to distinguish between attacks that originate within or outside a corporate network. In this study, the most significant features for enhancing the IDS efficiency and creating a smaller dataset in order to reduce the execution time for detecting attacks are selected from the sizeable network dataset. This research designed an anomaly-based detection, by adopting the modified Cuckoo Search Algorithm (CSA), called Mutation Cuckoo Fuzzy (MCF) for feature selection and Evolutionary Neural Network (ENN) for classification. The proposed search algorithm uses mutation to more accurately examine the search space, to allow candidates to escape local minima. Moreover, the value of the solution is evaluated based on the objective function and the Fuzzy C Means (FCM) clustering method used to provide the best results for the overlapping dataset and create the fuzzy membership search domain which includes all possible compromise solutions. A proposed model has been practically used to the problem of intrusion detection as well as been validated using the NSL-KDD dataset. The experimental results reveal that reducing features by selecting and utilizing the most relevant features can improve execution time and at the same time enhance the efficiency and performance of IDS.

INDEX TERMS Intrusion detection systems (IDSs), multilayer perceptron (MLP), multiverse optimization (MVO), cuckoo search algorithm (CSA), feature selection (FS), NSL-KDD dataset.

I. INTRODUCTION

In recent years, the number of smart device users is rapidly increasing. This steered to a substantial rise in network traffic. Hence this expansion has posed some security problems such as a variety of network attacks, whether known or unknown. Intrusions can be described as efforts or actions to threaten a computer or network in terms of its confidentiality, its integrity and/or its availability. Intrusion Detection System (IDS) is among the best methods for detecting attacks, as it involves a software or hardware system that tracks, evaluates and detects on-going activities from both inside and outside the network as an unauthorized activity [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

The huge data adversely affected the performance of IDS, also redundant and irrelevant information found within its traffic is responsible for poor performance in IDS [2], [3]. Over the years, several methods were proposed to highlight an issue associated with the IDS, amongst them are the Feature Selection (FS) methods and optimization techniques. A Feature selection technique is employed to prune high dimensions found in huge datasets by selecting the most relevant features to avoid curse dimensionality encountered during the construction of the IDS model. The selected features, a subset of the original dataset is used to simplify model construction in order to improve execution time by reducing training time. Thus, unnecessary features are filtered away [4]. The FS approach is made up of a wrapper, filter, and hybrid that with a large amount of data, complex calculations

are involved, and this invariably affects the efficiency of the FS approach. On the other hand, optimization algorithms employ a simple method that is inspired by nature for solving complex problems [5]. The optimization algorithms have been designed based on ideas inspired by nature involving a selection of the best option for certain given objectives. These kinds of algorithms can be grouped into three classes: Swarm-Based Algorithms (SBAs), Evolutionary Algorithms (EAs), and Trajectory-Based Algorithms (TBAs). This study proposes a new method for handling feature selection using Swarm Intelligence (SI) technique. The Swarm-Based Algorithm (SBA) is a popular algorithm for generating a relatively cheap, fast and vital alternative to the number of complicated issues. Instances of SBAs include Firefly Algorithm (FA), Particle Swarm Optimization (PSO), Cuckoo Search Algorithm (CSA), and the Artificial Bee Colony (ABC) [3]. Among the optimization strategies, the CSA has gotten a lot of researchers' attention compared with other techniques. This is because it is related to a small number of components required within the initial search. This makes it possible for even untrained users to interact with solutions based on the CSA without any difficulty. The meta-heuristic algorithm employed by CSA ensures a balance in its exploration and exploitation [6]. In order to improve the efficiency of CSA to achieve better solutions, a modified CSA is applied as a feature selection in this research. Furthermore, to detect intrusions, the NSL-KDD dataset will be used to evaluate the evolutionary neural network classifier. A comparison of the classifier's outcomes shows that our suggested model offers greater efficiency over the current intrusion detection systems. This article is organized as follows, section 2 presents the related research works on IDS, and the proposed method is presented in section 3. The dataset used for experimentation is presented in section 4 while section 5 presented the experimental setup and findings of this study. Finally, section 6 concludes the article.

II. RELATED WORK

Feature selection is among the important techniques that ensure effective anomaly-based detection. The procedure of selecting features eliminates irrelevant and redundant features, that produce appreciable impacts on improving IDS performance, especially in terms of dimensionality and execution time [7]. Also, a combination of ANN and Evolutionary Algorithm (EA) can produce an advanced technique to develop an efficient anomaly detection approach for IDS. Evolutionary Neural Network (ENN) algorithm is a form of neural network in which evolution is fundamental in the optimization of its learning process [8]. This section provides a brief discussion of some recent research including feature selection approaches and Evolutionary Neural Network in IDS that have motivated us in devising our approach.

Reference [9] proposed the Information Gain (IG) feature selection technique that combines the Support Vector Machine (SVM) with Bat Algorithm (BA) in order to improve IDS performance. Their model was evaluated using the NSL-KDD dataset where 26 out of 41 features were selected. The hybrid model, IG-BA-SVM obtained a high detection rate of 95.76 %, an accuracy rate of 94.16% and lower false alarm rate of 0.0408 compared to IG-SVM.

Reference [9] compared the detection accuracy of Multilayer Perceptron (MLP)-based IDS with chi-square, gain ration and information gain methods of feature selection. These are the feature ranking method which ranks the features of the dataset according to their importance in descending order. Out of 41 features of the NSL-KDD dataset, 25 were selected based on the ranking presented by each feature selection approach while the MLP is trained with this reduced dataset. It is discovered that among them chi-square achieved better results for accuracy and execution time of 78.57% and 74.78% respectively.

Reference [10] addressed the problem of dimensionality reduction in the intrusion detection problem areas. The suggested approach employs the Ant Colony Optimization (ACO) algorithm and the nearest neighbor as a classifier to select significant features for identifying new intrusions. The experiments are carried out and determined on the NSL-KDD dataset. The suggested approach has chosen 24 features out of 41 and achieved a 98.9% accuracy and 2.59 % false alarm rate. Findings show that the proposed model could be effective for IDS.

Reference [11] proposed a model to detect intrusions based on the Genetic Algorithm (GA) for selecting the best subset of features and Naive Bayes (NB) for classification purposes. A result obtained shows the potentials of the proposed model to enhance the precision and detection rate of intrusion detection with the least number of features. The proposed method (GA-NB) achieved DR, ACC and FAR of 97.63 %, 97.51% and 2.60 % respectively.

Reference [12] proposed a new FS technique for IDS to reduce the False Positive (FP) of existing IDSs and overcome low-performance problems. FS algorithm is described based on Random Forest Classifier (RFC) and Sequential Forward Floating Search (SFFS) to reduce system resource usage and improve classification accuracy. Experiments were conducted using the NSL_KDD dataset to establish the superiority of the (SFFS-RF) FS technique. The results obtained a 0.4% false alarm rate and 99.89% accuracy. This result shows that by decreasing the number of features, the classification performance of a model improves. Thus, SFFS-RF could have a positive impact on the classification accuracy and improve system resource management. In this way, it is possible to shorten the execution time by selectively significant features that affect intrusion detection. The detection model generated based on the FS technique could be used as a base model for a lightweight IDS.

Reference [13] applied the Binary Firefly Algorithm (BFA) in order to improve feature selection in IDS. The suggested approach was conducted with the use of the Naive Bayesian Classifier. The NSL-KDD dataset was utilized and provided empirical proof of improved randomization and movement of FA by calculating the distance using the hamming

distance approach. Due to the usage of the binary sequence instead of standard FA, this improvement can provide superior outcomes about efficiency and precision. The proposed method (BFA-NB) achieved 92.02% accuracy and a 7.98% false alarm rate with 14 features Out of 41.

Reference [14] the performance of IDS is enhanced by employing the information gain feature selection technique and SVM with Particle Swarm Optimization (PSO) for classification. The result obtained indicates that the superiority of the IG-PSO-SVM detection model with a 0.9% false alarm rate and 99.8% accuracy is high reliability in intrusion detections on the NSL_KDD dataset.

Reference [15] presented an IDS that utilizes a backpropagation MLP neural network. In the scheme of their intrusion detection system, the neural network weights have been optimized using the GA algorithm. They employed different types of attacks, which were based on the KDD-Cup99 dataset. The improved genetic backpropagation obtained DR and FAR of 91.34 % and 26.02% respectively. The proposed method recorded a higher detection rate and a lower false alarm rate compares to the traditional MLP results.

Reference [16] designed the IDS model using a modified PSO to optimize a backpropagation neural network. They exploited the benefits of universal search capacity to modify PSO. The efficiency of the proposed IDS model was evaluated using the KDDCup99 dataset. The proposed method recorded a higher detection rate and a lower false alarm rate compared to the traditional Back Propagation (BP) results.

Reference [17] a novel IDS model is developed on the basis of the BPNN. It is a new approach investigated using the proposed method for IDS which is based on D-S theory and GABP. Empirical outcomes with the KDDCup99 dataset attained an acceptable detection rate of 97.5 % and a low false-positive rate of 2.1%. Results proved that the proposed method is an effective method for IDS.

Reference [18] proposed a new evolutionary neural network using a multiverse optimizer and artificial neural network-multilayer perceptron which has improved the performance of IDS to detect new forms of attacks. They evaluated the model using two datasets one of which is used in this study; it is known as the NSL-KDD dataset. The experiment proves that the proposed method recorded a significant improvement in accuracy, detection rate and false alarm rate of 98.21%, 96.25%, and 0.032 % respectively.

Reference [19] proposed a training algorithm with its basis in the newly suggested Whale Optimization Algorithm (WOA). This algorithm has been shown to be capable of solving numerous optimization issues and outperforms the existing algorithms. The preference of WOA over training MLPs is due to its fast speed in converging and the method employed for avoiding local optima. The WOA is used for identifying the best values for weights and biases in order to reduce the MSE. This algorithm demonstrated better performance compared to the existing methods in terms of accuracy and convergence. Reference [20] addresses feed forward NN training issues employing a recent metaheuristic ''Locust Swarm Optimization'' (LSO) algorithm to enhance advanced detection systems. LSO enables ANN to develop its weights and adjust its factors to prevent being trap in local minima such that it could circumvent under fitting or over-fitting. Tests carried out have demonstrated that the suggested method is efficient in training ANN compared to more commonly known training approaches, like PSO and GA. UNSW-NB15 and NSL-KDD datasets were used for evaluating the success of the proposed approach. The proposed approach allows alternative IDS solutions and increases the detection rate.

Given that both feature selection and evolutionary neural network methods improve the performance of the intrusion detection system a combination of these two approaches has been used in this research to create an efficient anomaly-based intrusion detection system.

III. PROPOSED METHOD

One of the important problems encountered in many algorithms used for feature selection is the high computational time complexity. For this reason, in this study, the cuckoo search algorithm was used to select the features due to the main benefit of the CSA compared to other algorithms which are the number of parameters needed to be configured in the initial search is very small and can deal with multimodal problems naturally and efficiently.

This study implements and designs an anomaly-based detection system with a new feature selection method using a modified cuckoo search algorithm and a combination of MVO-ANN for classification. A new feature selection method, called Mutation Cuckoo Fuzzy (MCF) which combines mutation operator with cuckoo search and Fuzzy C Means (FCM) clustering is proposed. The proposed feature selection method increases the efficiency of CS algorithm to select the best features.

In addition, the distinction between normal behavior and attack is very important, so classification plays an important role in IDS. Among machine learning techniques, one of the most widely used methods for intrusion detection systems is an artificial neural network. Traditional and conventional classification techniques require a good understanding of the underlying assumptions of the probability model. In contrast, ANNs have many advantages including capable of adapting to the unknown and underlying system model, flexibility which can learn complex relationships within the dataset without making assumptions about the specific nature of any relationship. Furthermore, having self-organization and abilities to learn what makes them particularly useful in the prediction and classification of internet traffic, especially in high dimensional datasets and overcoming the difficulties in model buildings. However, one of the main challenges in the implementation of an Artificial Neural Network- Multilayer Perceptron (ANN-MLP) is the training part using Backpropagation (BP) algorithm. Because BPANN is likely to get into local structural minima, which negatively affects the

capability of accurately assigning the ANN structures [18]. To overcome ANN-MLP limitation to avoid getting into local minima, Multiverse Optimizer (MVO) is used to train ANN. Multiverse optimizer is one of the bio-inspired evolutionary programming algorithms, recently proposed [21]. The objective of the MVO algorithm is to locate the global solution for the given optimization problem. Results of MVO compared with some other algorithms prove that the proposed algorithm is able to provide very competitive results and outperforms the best algorithms such as Grey Wolf Optimizer, Particle Swarm Optimization, Genetic Algorithm, and so on [22]. For that, in this research MVO is used for training ANN.

The proposed anomaly-based detection is categorized into three modules, as follows:

Step 1: In the first module, using MCF, efficient and useful features in terms of accuracy are separated out of 41 features in the dataset as a set of optimal features.

Step 2: In the second module, selected features are sent to the ANN module as a training characteristic of the input information dimension. This module is planned to be a Multilayer Perceptron (MLP) with a single input layer, one hidden layer, and one output layer architecture. This training is carried out by transferring the weights to the Multiverse Optimizer module. After each iteration, the MVO is employed to update synaptic weights. In this module, a set of weights is integrated into an ANN for each iteration of the training method. Then, the final fitness values are returned for each individual performance, which is measured based on the training dataset. The Mean Square Error (MSE) is selected for the suggested MVO training algorithm as a known fitness function in this research. The synaptic weights are derived by decreasing the MSE value such that the process of training terminates whenever the required maximum iterations are attained, thereafter, an update on the knowledge base is performed.

Step 3: The third module involves the predicted output. In this step, inputs are conducted into the trained ANN from the testing dataset in order to estimate the output. Therefore, the ANN testing process can be regarded as the most closely matched to any of the target classes according to the expected output.

A. CUCKOO SEARCH ALGORITHM (CSA)

CSA is a metaheuristic algorithm that gets its motivation from nature and created by [23] in 2009.

The algorithm (CS) mimics the natural behavior of cuckoos, the ‘‘obligate brood parasitism’’ of certain cuckoo species that lay its eggs in the other host birds’ nests. Researchers have used three rules to define the CSA in order to make it implementable as a computer algorithm:

- Each cuckoo at a time lay one egg and dumps it randomly in a chosen nest.
- The optimal nests containing excellent quality eggs and they will be passed on to the following generation.
- There are several existing host nests which are been predetermined, and the egg laid by a cuckoo could be identified with a probability of $P_a \in (0,1)$. When this occurs, the egg is either removed or the nest is abandoned and a new one is constructed.

Regarding the mentioned rules, the CS is implemented in the following manner:

Individual eggs in a nest signify candidate solutions. Thus, a cuckoo can lay just a single egg in a nest while each nest can contain several eggs of solutions, generally. The CS is responsible for generating novel and possibly superior solutions that will be replacements for the inappropriate solutions in the ongoing population. The quality of the solutions is assessed and solved with the objective function of the problem that needs to be maximized. The last estimation rule is known by P_a as the "probability of switching", which determines when the worst host nest is to be replaced with a new randomly generated nest. This factor provides the balance of two parts of the CS process, exploration and exploitation [24]. Due to this, an excessive amount of exploitation induces early convergence, whereas too much exploration slows down the convergence. In the generation of new solutions x (t+1) for cuckoo i, Equation 1 is employed to conduct Levy flight.

$$x_i\,(t+1) = x_i\,(t) + \alpha \oplus Levy(\lambda) \tag{1}$$

where: $\alpha > 0$ denote the step size is assigned based on the scales of the problem. Often time, $\alpha = 1$ can be used. The symbol $\oplus$ is an entry-wise multiplication. Frequently, Levy flights arrange for a random walk however, their random steps are derived from a Levy distribution for large steps provided in Equation 2 which possesses an infinite mean with infinite variance.

$$Levy \backsim u = t^{-\lambda} \tag{2}$$

Based on the three rules specified earlier, the essential step of the Cuckoo search (CS) technique is shown in algorithm 1.

Algorithm 1 Cuckoo Search Algorithm (CSA)

```
begin
    Objective function f(x), x = (x_1, ..., x_d)^T
    Generate initial population of n host nests
    x_i(i = 1, 2, ..., n)
    while (t < MaxGeneration) or (stop criterion)
        Get a cuckoo randomly by Levy flights
        evaluate its quality/fitness F_i
        Choose a nest among n randomly
        if (F_i > F_j),
            replace j by the new solution.
        end if
        A fraction (p_a) of worse nests are abandoned and
        new
        ones are built.
        Keep the best solutions.
        Rank the solutions and find the current best
    end while
    Post process results and visualization
end
```

B. CUCKOO SEARCH WITH FUZZY C MEANS

Considering that CS is used to generate new and potential solutions, the objective function plays an important role in evaluating solutions and replacing them with other existing solutions. The fitness function (evaluation function) denotes how a given solution is closer to the final solution of the desired problem. It determines how fit the solution is with reference to the problem under consideration [25]. For this purpose, due to the benefits of FCM clustering, we have used it as an objective function in this research. According to FCM, the dataset x must be partitioned into C clusters based on the characteristics of the dataset. The data has to be converted into fuzzy. A membership function $\bar{\mu}_i(x_j)$ for the fuzzy representation is defined by Equation 3 where i = 1, 2 . . . , n and $j = 1, 2 \ldots$, d.

$$\bar{\mu}_i(x_j) = \frac{x_{ij} - min(x_i)}{max(x_j) - min(x_j)} \tag{3}$$

An extensively utilized objective function for fuzzy c means clustering is the weighted within-groups sum of squared errors J_m, which is employed for defining the restricted optimization issue as discussed in Equation 4:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\| \tag{4}$$

where: $1 \leq m \leq \infty$, m is any real number higher than 1, u_{ij} is the extent of membership of x_i in the cluster j, x_i is the i^{th} component of d-dimensional recorded data, c_j is the center of the cluster, and $|| * ||$ is any norm stating the resemblance between any measured data and the center. The partitioning of fuzzy is performed via a repetitive optimization of the objective function, through the update of membership u_{ij} and the cluster centers c_j by Equations 5 and 6 respectively:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_j\|} \right)^{\frac{2}{m-1}}} \tag{5}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{6}$$

The objective of clustering optimization is utilized to locate the best cluster centroid with the assistance of the iterative procedure. FCM as an effective objective function has been used in cuckoo search fitness function that can result in enhanced outcomes in locating the optimal centroid and evaluate the fitness of nest found in the population matching the random number. First, Cuckoo's algorithm chooses several features among all of them. Then, to determine the desirability of each one, the FCM algorithm is used. The process of using FCM in cuckoo search presents in Fig.1.

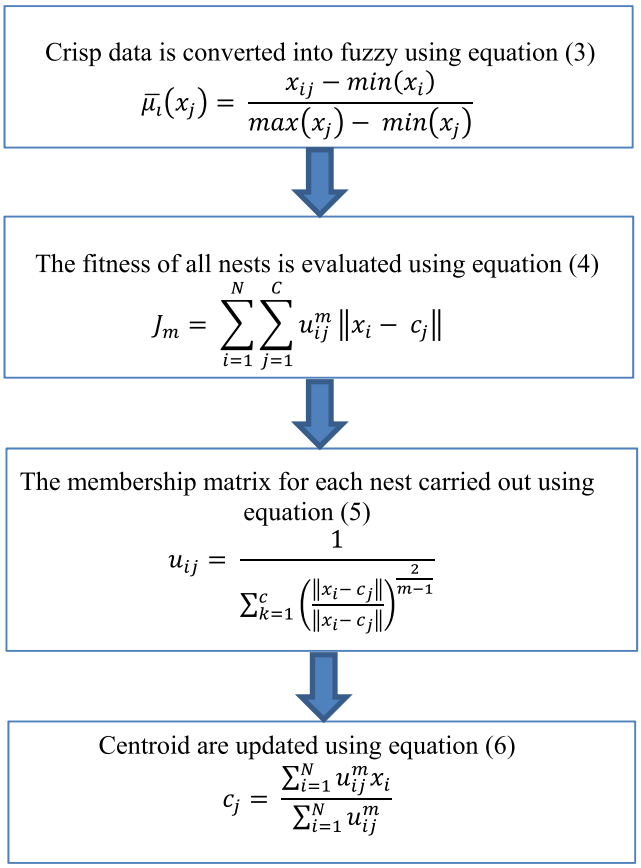

FIGURE 1. The process of using FCM in CS.

C. MUTATION CUCKOO SEARCH ALGORITHM

The CS algorithm examines a small search space due to fast convergence in some cases.

The method should increase the search space to be ensured about the convergence and attains optimal response [24]. Consequently, in this research, the genetic algorithm mutation operator is used to create more space and more varied solutions. For this purpose, two steps are considered for a combination of mutation with cuckoo search algorithm:

- Random selection of a nest
- Random selection of an egg

The traditional CS considers a single egg each time in a nest using Levy flight. The mutation is defined as "a genetic operator that changes one or more gene values in a chromosome in the genetic algorithm from its initial state." This can lead to the addition of a completely new gene value to the gene pool. The mutation is also defined as "an essential component of genetic searching because it assists in preventing the population from being trapped at any local optima. As indicated earlier, the cuckoo's eggs will impersonate the host bird's eggs. To explain this behavior, a mutation operator is included in the algorithm to provide a reflection of the mutation behavior of the cuckoo eggs' genes to enhance their productivity. Hence, using this strategy retains high-quality eggs whereas the lower quality eggs are rejected. Regarding the mutation process, when the new randomly selected cuckoo egg is better than the old one, the new one will replace the old cuckoo egg. This approach ensures that in the next generation the best candidate solution will always be maintained. In order to enrich the population's diversity, it is randomly selected

the nest for mutation, as discussed in Equation 7.

$$M_i = X_i + r\,(X_{best} - X_{worst}) \tag{7}$$

where X_i represents the i-th nest's position that X_{best} and X_{worst} indicate the best and worst individual in generation population. Similarly, r is generated in the range from 0 to 1. Using mutation combines the best and worst individual to ensure the diversity of the population. If the mutation's individual is promoted, it will replace the current individual so that new cuckoo's population $x=(x_1, \ldots, x_d)^T$ is generated.

For performance evaluation, the accuracy of the selected attributes is measured. Modified Cuckoo search algorithm using mutation in conjunction with a fuzzy c means can solve the drawbacks of the traditional cuckoo search as having the better result for evaluation of the quality of a feature for feature subset selection. The general pseudo-code of the proposed feature selection called Mutation Cuckoo Fuzzy (MCF) is presented in Algorithm 2. The process of feature selection starts with the creation and initialization of the parameters such as generation, count nest, P_α and get a cuckoo randomly by levy flights. Then calculate error function using fuzzy c means clustering. After that, create the nests and choose the best one between them. Place the egg in the created nest. Calculate P_α and change nest randomly using mutation. Afterward, keep the best solution and show the best nest based on error value and accuracy of the classification. After applying feature selection on the NSL-KDD dataset, 22 features are selected out of 41 features. These selected features are used as input for the classification part performed using MVO-ANN.

D. ARTIFICIAL NEURAL NETWORK (ANN)

In this research, in terms of classifying data as an attack and normal in anomaly-based detection, the well-known algorithm, Multilayer Perceptron (MLP) has been chosen with binary classification and one hidden layer. The ANN includes extremely interconnected parallel processing components and distributes several inputs to a set of expected outputs. ANN-MLP classification is supervised learning with known class labels [26]. The MLP is a class of feedforward-an artificial neural network that includes at least three layers: input, output and hidden layer. Normally, each input is multiplied by the matching weight to the network, and the total of them makes a weighted sum function; then the result of the sum function is passed via a transfer or activation function. Fig. 2 presents the simple structure of the ANN.

Algorithm 2 The general pseudo-code of the proposed feature selection MCF

```
begin
    Objective function f(x), x = (x_1, ..., x_d)^T
    Generate initial population of n host nests
    x_i(i = 1, 2, ..., n)
    while (t < MaxGeneration) or (stop criterion)
        Get a cuckoo randomly by Levy flights
        set fuzzy cluster error as fitness function
        evaluate its quality/fitness using FCM
        Choose a nest among n randomly
        if (F_i > F_j),
            replace j by the new solution.
        end if
        calculate (p_a) parameter as a worst nest
        Randomly selected the worst nests for mutation
        Keep the best solutions
        Rank the solutions and find the current best
    end while
    Post process results and visualization
end
```

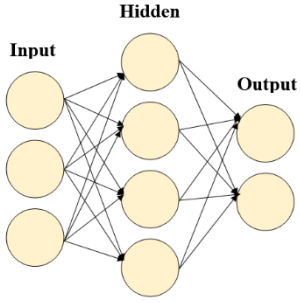

FIGURE 2. Simple architecture of the ANN.

The total function is computed as the addition of the products, the initial weights and additional bias, as shown in Equation 8.

$$Y = \sum_{i=1}^{k} w_{ij} * I + \beta_j \tag{8}$$

Back Propagation (BP) is the familiar training algorithm used for training ANN in supervised mode. The presence of numerous factors in the structure of ANN is the major problem of implementing the system. In particular, there is the possibility of BPANN getting into the local minima which will have a negative impact on the ability to assign ANN structure correctly [18]. For this reason, in this research instead of using the backpropagation algorithm, Multiverse Optimizer (MVO) is applied in ANN to adjust weights and minimizing the error function.

E. MULTIVERSE OPTIMIZER (MVO)

The MVO algorithm consists of three cosmological perceptions: white holes, black holes, and wormholes which are parts that form the universe. These three conceptions constitute the idea behind the MVO algorithm, that provides the simulation of the dynamics and universes interaction via black holes, white holes, and wormhole [21]. In MVO, the inflation rate is corresponding to the fitness while the term "time" parallels the iteration. The rules itemize below is applicable to any optimization process.

- A higher inflation rate has a greater probability of an existing white hole.

- A lower inflation rate is greater probability that a black hole exists.
- The best universe is the result of objects randomly moving through wormholes.

When objects between universes are exchanged, a universe possessing an object is sending to the other universes from a higher inflation rate to the lower inflation rate.

Furthermore, a universe with lower expansion rate gets extra objects from better universe so that it will attain a steady status in order to become an optimal universe with enhanced inflation rate. Amid optimization process, each universe is aligned with its inflation rates, and the selected universe is identified by employing the roulette wheel to be the white hole.

The Traveling Distance Rate (TDR) and the Wormhole Existence Probability (WEP) are the two major coefficients; lb_j represent the lowest j^{th} variable, ub_j denotes the highest j^{th} variable [18]. The formulas for the two coefficients are given by 9 and 10 respectively:

$$TDR = 1 - \left(\frac{l^{\frac{1}{p}}}{L^{\frac{1}{p}}} \right) \tag{9}$$

$$WEP = min - l * \left(\frac{min - max}{L} \right) \tag{10}$$

where p represents the exploitation factor; and l is the existing iteration, L indicates the highest number of iterations. WEP and TDR are elevated at all iterations to attain greater accuracy around the best-gained universe in exploring/local search. The general steps of the MVO algorithm are shown as follows:

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 the inflation rate (fitness) for each universe.
4) Calculate WEP, TDR by equations 9,10.
5) Exchange objects between universes.
6) Objects in each universe teleport to the best universe.
7) Go to step 3 if the end criterion is not satisfied.
8) Returned the best universe formed thus far.

The optimization process commences with the creation and initialization of the factors, such as population size and the upper and lower bounds. Afterward, universes are randomly initialized as a set based on lower and the upper bounds. A corresponding fitness values are computed for individual universes so as to describe the optimal potential inflation rate. Then, at individual iterations, high inflation rates objects in the universes tend to migrate to the universes containing low inflation rates through the white or black hole. Consequently, objects in individual universes move randomly into the optimal universe via wormholes. Ultimately, the optima universe is formed at the completion of the operation.

F. TRAINING OF ANN WITH THE MVO ALGORITHM

The goals of optimizing ANN with training method are to locate the ANN synaptic weights and reduce the MSE representing the cost of ANN function. Through the generation of population of solutions, the MVO algorithm initializes the optimization process on the assumption that every universe is considered as an individual in the randomly generated solution population. The measurement of the issue connotes the size of the solution. MVO individual's depiction and design are significant considerations in ANN's training. Each individual in the ANN training reflects all the ANN structure's weights and biases. The training method aims at finding the right values, reduce the error and achieving the greatest accuracy of classification. A pseudo-code of the MVO-ANN process is shown in Algorithm 3.

Algorithm 3 MVO-ANN training pseudo-code

```
begin
    Initialize the training parameters: WEP, TDR, lb, ub, Max-
    iteration
    Create a set of random individualize based on the
    problem dimension
    for each individual do
        Calculate the MSE for the individual
        if the current MSE < the global minimal MSE
            Update the global minimal MSE
        end if
    end for
    for iteration (t) <= Max-iteration do
        for each individual do
            Calculate the parameter of MVO: WEP,
            TDR
            Run the optimization process
            Exchange objects between the universes
            Objects to each universe teleport to do the
            best
            universe
        end for
        for each individual do
            Calculate the MSE for individual
            if the new current MSE < minimal MSE
                Update the globally minimal MSE
            end if
        end for
        Save the current best solution with the minimal
        MSE
    end for
    return the best solution of the minimal MSE
end
```

All individuals in MVO are vectors that comprise connecting weights between ANN layers. The number of objects for each individual is computed as presented in Equation 11:

$$Indv_{nbr} = (n * m) + (2 * m) + 1 \tag{11}$$

MSE is been used primarily as cost function in a proposed MVO training algorithm. An MSE can be computed using

Equation 12:

$$MSE = \left(\frac{1}{T_n}\right) * \sum_{i=1}^{T_n} (output - input)^2 \qquad (12)$$

Here, the real data is the input while the approximated values is the output and T_n is the frequency of amounts in dataset.

IV. DATASET NSL-KDD

The efficiency of the proposed anomaly-based detection using MCF feature selection and MVO-ANN classification was examined, and its performance was compared with some other current methods in the same area.

In order to validate, well known available dataset namely NSL-KDD was used. This dataset is dedicated to offline IDS evaluation. Table 1 shows the features in the NSL-KDD dataset. The KDD Cup 99 is the most popularly used benchmark dataset for evaluating IDSs that are derived from DARPA 98 dataset. In the training dataset, there is nearly 5 million overall number of connection records, but they suffer from some unnecessary elements in the testing and training sets. These redundant elements have a significant effect on the performance of the system and lead to an unsatisfactory assessment of the IDSs [27].

Hence, the NSL-KDD dataset is structured to improve the KDD Cup 99 data and address the characteristic issues of the last mentioned. The dataset is derived from the various portions of the original KDD Cup 99 dataset, with no unnecessary elements and repetitions. Additionally, in order to enhance the precision of the IDS assessment, the issue of unbalanced dissemination in the testing and training set was resolved.

TABLE 1. Features in the NSL-KDD dataset.

Type	Features
Nominal	Protocol_type(2), Service(3), Flag(4)
Binary	Land(7), logged_in(12), root_shell(14), su_attempted(15), is_host_login(21),, is_guest_login(22)
Numeric	Duration(1), src_bytes(5), dst_bytes(6), wrong_fragment(8), urgent(9), hot(10), num_failed_logins(11), num_compromised(13), num_root(16), num_file_creations(17), num_shells(18), num_access_files(19), num_outbound_cmds(20), count(23) srv_count(24), serror_rate(25), srv_serror_rate(26), rerror_rate(27), srv_rerror_rate(28), same_srv_rate(29) diff_srv_rate(30), srv_diff_host_rate(31), dst_host_count(32), dst_host_srv_count(33), dst_host_same_srv_rate(34), dst_host_diff_srv_rate(35), dst_host_same_src_port_rate(36), dst_host_srv_diff_host_rate(37), dst_host_serror_rate(38), dst_host_srv_serror_rate(39), dst_host_rerror_rate(40), dst_host_srv_rerror_rate(41)

The NSL-KDD data set has the following advantages over the original KDD data set [28]:

- It does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records.
- There are no duplicate records in the proposed test sets; therefore, the performance of the learners is not biased by the methods which have better detection rates on the frequent records.
- The number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD data set. As a result, the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques.
- The number of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion.
- The NSL-KDD dataset comprises 41 features and 42nd attribute labeled as ''normal connections'' or ''attack types'' and has four attack classes that are probe, DoS, U2R, and R2L [29].

V. EXPERIMENTAL SETUP AND RESULTS

Findings and discussions about the efficiency of the proposed anomaly intrusion detection method are presented in this section. The massive amount of irrelevant and redundant features causes slow training data and high execution time. Consequently, to address this issue, many forms of IDSs with various feature selection techniques have been suggested. In this research, an efficient anomaly intrusion detection method with a new suggested feature selection and the evolutionary neural network has been proposed to offer possible ways for improving the performance of IDS and reduce the execution time.

A. EXPERIMENTAL SETUP

Implementation and evaluation of the suggested model were performed on a Personal Computer (PC) with Core I5 3.20 GHz CPU and 8 GB RAM in MATLAB 2017a. Due to the many advantages that MATLAB software has, as described below, it has been used in this research [30].

- Complex mathematical operations like matrix multiplication and addition can be easily accomplished in a single code by using MATLAB.
- Data can be saved in variables with very simple commands which are easy to use by storing numbers in a vector or matrix where no use of loops is needed.

- MATLAB's functionality can be greatly expanded by the addition of toolboxes. These are sets of specific functions that provided more specialized functionality.

We tried to vary the number of host nests (population size n) and the probability of discovery P_a. We have used different settings for n (5, 10, 15,20, 25,30,50) and for P_α(0, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.4, 0.5). From test phase simulations, we found that n = 30 and $P_\alpha = 0.25$ are sufficient for most optimization problems. Therefore, we used fixed n = 30 and $P_\alpha = 0.25$ and maximum number of iterations =200 to reach the optimum based on cuckoo search method. Furthermore, given the suitable parameter settings to significantly influence the model performance, common settings are applied to the MVO parameter. Minimum and maximum WEP are set to 0.2 and 0.6 respectively. Also the number of individuals and iterations were set as 4 and 100 respectively after testing the model with different number of hidden layer (4, 5, 10, 30) and iteration (100, 200, 250) to get optimum result. The scale [0, 1] are assigned for all inputs in the experiment. The proposed IDS evaluated based on several factors, including the accuracy (ACC), false alarm rate (FAR) and detection rate (DR). The main instance for evaluation contains True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP) that are encompassed in a Confusion Matrix (CM) which has a dimension of NN and presents a classification result. The abbreviations of the CM are presented in Table 2.

TABLE 2. The abbreviations of the confusion matrix.

TP	Number of connections successfully classified as anomalies by the classifier
FN	Number of anomalies connections that are misclassified as normal by the classifier
FP	Number of normal connections that are misclassified as anomalies by the classifier
TN	Number of normal connections that are successfully classified as normal by the classifier

CM is a 2×2 matrix where the columns values correspond to the expected classes and while the rows denote actual classes as shown in Table 3.

To compute the accuracy, false alarm Rate, and detection rate, the following equations applied.

Accuracy (ACC) or Classification rate is the ratio of correctly classified instances divided by the total number of instances.

$$ACC = \frac{Correctly\ classified\ instances}{Total\ number\ of\ instances}$$
$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

TABLE 3. Confusion matrix.

Actual	Predicted Attack	Predicted Normal
Attack	**TP**	**FN**
Normal	**FP**	**TN**
Total	**TP+FP**	**FN+TN**

Detection Rate (DR) is the ratio of the number of correctly detected attacks and the overall number of attacks.

$$DR = \frac{Correctly\ detected\ attacks}{Total\ number\ of\ attacks}$$
$$DR = \frac{TP}{TP + FN} \quad (14)$$

False Positive Rate (FPR): calculated by finding ratio between number of normal instances detected as attack and overall number of normal instances.

$$FPR = \frac{Number\ of\ normal\ instances\ detected\ as\ attacks}{Totalnumber\ of\ normal\ instances}$$
$$FPR = \frac{FP}{FP + TN} \quad (15)$$

B. RESULTS AND DISCUSSION

41 features are taken into concern in evaluating the set of data in order to detect attacks. Out of these 41features, 22 are selected through a proposed feature selection technique as presented in Table 4.

The selected features are categorized into three priorities based on importance of features in terms of accuracy.

These features determined the significance level of accuracy and the best results obtained is shown in Fig 3.

Considering that only showing the algorithm predicts well is not enough. It is important to attribute the predictions to the elements that contribute to accuracy. Because in this research the aim of feature selection is to minimize the number of features, such that only the best features are retained in the subset, whilst maximizing the classification accuracy. In this reason, the importance of each feature was estimated to explain the predictive power of the selected features and each feature was evaluated independently of each other in the proposed algorithm to gain importance of each of the features individually. According to the evaluation of each feature, the first category with 10 features has highest importance with topmost significance, named "First Priority" which has obtained 97.73% accuracy rate.

Although selective features in the second and third category are less impact, they are effective in enhancing accuracy. By combination of first priority and second priority the accuracy obtained was 98.65% which shows second priority group improved the accuracy about 0.92 % and with all three groups the accuracy produced was 98.81% that with third priority group accuracy was improved by 0.16% as shown in Table 5 also accuracy for each priority presents in Fig 4.

The performance of this method in terms of accuracy, detection rate and false alarm rate are measured in two experiments. Experiment I, all the features are used (41 features of

TABLE 4. Selected features from NSL-KDD dataset.

No	Name	Description
1	Duration (1)	Length of time duration of the connection
2	Protocol type (2)	Protocol used in the connection
3	Flag (4)	Normal or error status of the connection
4	Src_bytes (5)	Number of data bytes from source to destination
5	Land (7)	1 if connection is from/to the same host/port; 0 otherwise
6	Wrong_fragment (8)	Number of "wrong" fragments
7	Urgent (9)	Number of urgent packets
8	Logged_in(12)	1 if successfully logged in; 0 otherwise
9	Su_attempted(15)	1 if "su root" command attempted; 0 otherwise
10	Num_root(16)	Number of "root" accesses
11	Num_outbound_cmds (20)	Number of outbound commands in an ftp session
12	Is_host_login (21)	1 if the login belongs to the "hot" list; 0 otherwise
13	Is_guest_login (22)	1 if the login is a "guest "login; 0 otherwise
14	Count (23)	Number of connections to the same host as the current connection in the past two seconds
15	Serror_rate (25)	The percentage of connections among the connections aggregated in count
16	Srv_rerror_rate (28)	The percentage of connections among the connections aggregated in srv_count
17	Srv_diff_host_rate (31)	different destination machines among the connections aggregated in srv_ count
18	Dst_host_count (32)	Destination host count
19	Dst_host_srv_count (33)	Service count for destination host
20	Dst_host_srv_diff_host_rate (37)	Different host rate for destination host
21	Dst_host_rerror_rate (40)	R-error rate for destination host
22	Dst_host_srv_rerror_rate (41)	Srv-rerror for destination host

TABLE 5. Accuracy of results for selected features.

Selected Features	*Accuracy*
All 22 selected features	98.81
All selected features except third priority	98.65
All selected features except second and third priority	97.73

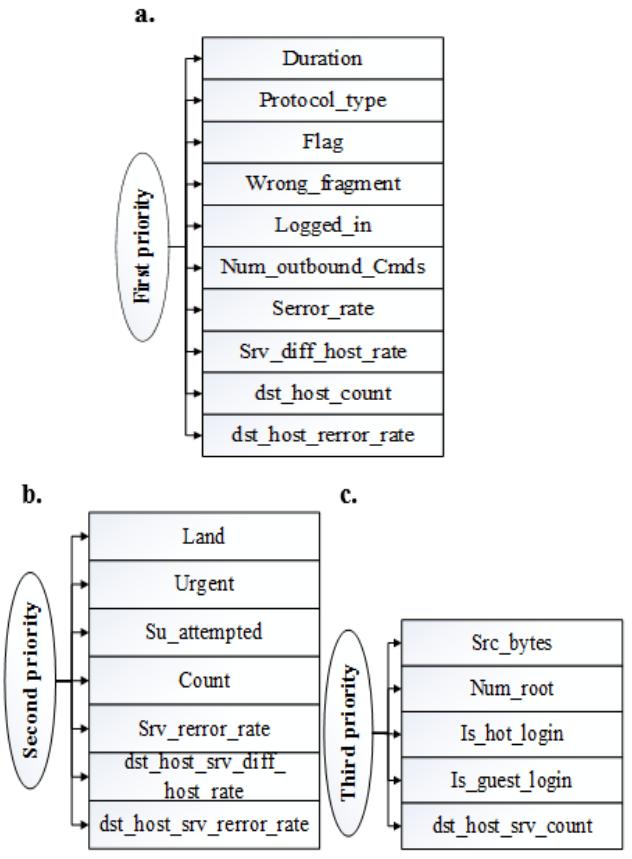

FIGURE 3. A. First priority, B. second priority and C. third priority.

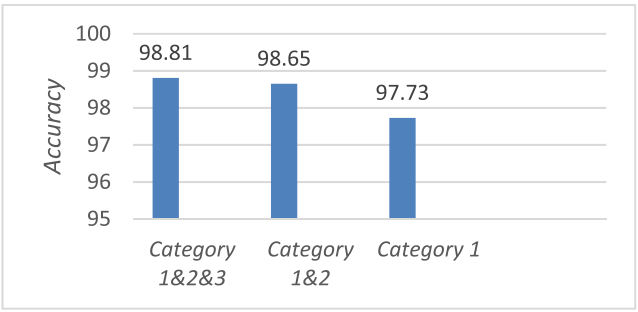

FIGURE 4. Accuracy for each priority.

NSL-KDD dataset) while experiment II only selected features are used (22 features selected out of 41 features in NSL-KDD dataset). The results are shown in Figures 5, 6 and 7.

Comparing the results in experiment I, that used all features (MVO-ANN) with experiment II, which utilized only significant features (MCF&MVO-ANN) demonstrate significant improvement in experiment II using the proposed method in terms of ACC and DR. In addition, the FAR has been greatly decreased compared to experiment I.

An anomaly intrusion detection system in the training phase builds a model of the normal network traffic to recognize acceptable behavior. For that training part is an important point for execution time. The smaller data size helps reduce training time, so feature selection can have a considerable impact. Hence, due to the reduction of features in this section, we have examined its effect on execution time.

The amount of time required from start to the event is referred to as execution time. In other word, it is the amount of time spend for computation part. In this research the calculation of execution time is performed using elapsed time method in MATLAB.

The measurement is divided into tic and toc functions. First, the toc function detects the elapsed time from the stopwatch timer which is triggered by the tic function. The tic function gets an internal time at the execution of toc

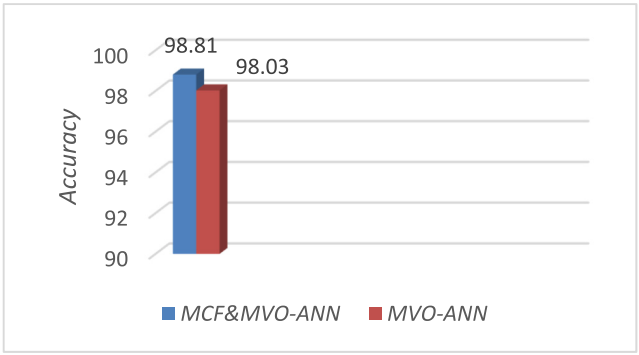

FIGURE 5. Comparison between the proposed method (MCF&MVO-ANN) and (MVO-ANN) in terms of accuracy.

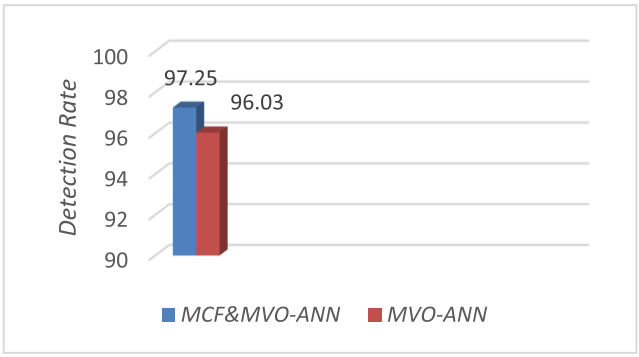

FIGURE 6. Comparison between the proposed method (MCF&MVO-ANN) and (MVO-ANN) in terms of accuracy.

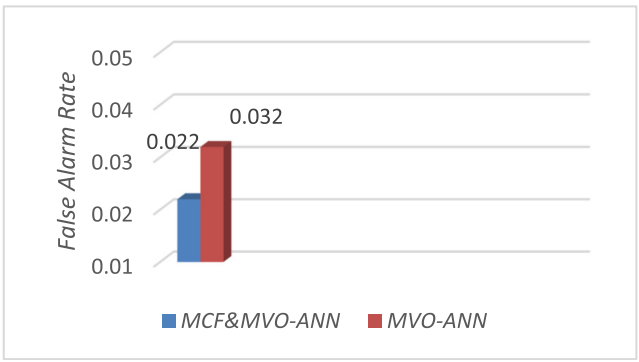

FIGURE 7. Comparison between the proposed method and MVO-ANN in terms of false alarm rate.

command, thus, displayed as the elapsed time since the most recent call to the tic function did not return the execution time yet in seconds. The result obtained for proposed MCF& MVO-ANN in execution time is 60.33s. However, execution time for MVO-ANN with all features (without feature selection) is about 163.07s as presented in Fig 8.

The result in Table 6 shows that reducing the number of feature and the use of important features only instead of all features has a great impact on execution time as well as improve the performance of the IDS model. The performance the proposed model (MCF&MVO-ANN) against the state-of-the-art baseline (MVO-ANN) on NSL-KDD dataset indicates that the proposed model outperforms the baseline in terms of ET, ACC, DR, FAR.

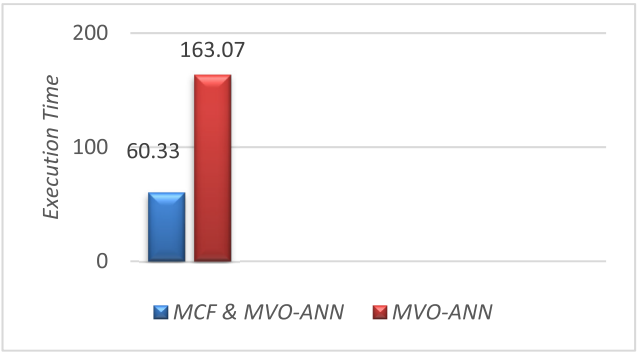

FIGURE 8. Comparison between the proposed method and MVO-ANN in terms of execution time.

TABLE 6. Comparison between the proposed method MCF&MVO-ANN and MVO-ANN.

Method	ET	ACC	DR	FAR
MCF&MVO-ANN	60.33s	98.81	97.25	0.022
MVO-ANN	163.07s	98.03	96.03	0.032

ET: Execution Time, ACC: Accuracy, DR: Detection Rate, FAR: False Alarm Rate.

TABLE 7. Comparison between the proposed model and other models using NSL-KDD dataset.

No	Author/ Year	IDS Model	ACC	DR	FAR	ET (in sec)	Number of features
1	[31]	IG-BA-SVM	94.16%	95.76%	0.04%	N/A	26 out of 41
2	[9]	Chi-square-MLP	78.57%	N/A	N/A	74.78	25 out of 41
3	[9]	Gain Ration-MLP	74.29%	N/A	N/A	133.05	25 out of 41
4	[9]	Information Gain-MLP	78.39%	N/A	N/A	109.33	25 out of 41
5	[10]	ACO-NN	98.9%	N/A	2.59%	N/A	24 out of 41
6	[11]	GA-NB	97.51%	97.63%	2.60%	N/A	10 out of 41
7	[11]	IG-NB	97.12%	96.19%	2.07%	N/A	39 out of 41
8	[12]	SFFS-RF	99.89%	N/A	0.4%	N/A	10 out of 41
9	[13]	BFA-NB	92.02%	N/A	7.98%	N/A	14 out of 41
10	[13]	BPSO-NB	90.63%	N/A	9.37%	N/A	22 out of 41
11	[14]	IG-PSO-SVM	99.8%	N/A	0.9%	N/A	20 out of 41
12	**Proposed Method**	**MCF& MVO-ANN**	98.81%	97.25%	0.022 %	60.33	22 out of 41

The comparison of the performance results between the proposed model (MCF&MVO-ANN) and some current researches using NSL-KDD dataset are presented in Table 7.

The proposed model (MCF&MVO-ANN) recorded a substantial performance in terms of DR and FPR over other models.

TABLE 8. Performance average results ofMCF&MVO-ANN.

Proposed Method	Average ACC	Average DR	Average FAR
MCF&MVO-ANN	98.16	96.83	0.029

The results show the potentials of the proposed method for developing practical IDSs.

Due to the structure of the ANN, there are random values with different starting points per each simulation during the training phase that cause different results for each running time.

There is no rule of thumbs for the number of running code to get an acceptable result. For this reason, by referring to the previous researches in this area the code was running for 30 times to ensure that it provides accurate results.

To get the average result, the mean calculation was used. The mean is the most commonly used mathematical measure of average where $\{x_1, x_2, x_3, \ldots, x_N\}$ are the observed values of the sample items x $^{-}$ is the mean value and N is the number of observations in the sample. The following Equation shows the calculation of mean value:

$$\text{Mean value} = \frac{\sum x_i}{N} \tag{16}$$

The following results shown in Table 8 are the performance average results of(MCF&MVO-ANN) achieved by calculating the mean value for ACC, DR and FAR.

VI. CONCLUSION AND SUMMARY

In this study, an anomaly-based detection system using new proposed feature selection called Mutation Cuckoo Fuzzy (MCF) is proposed to select the best feature subsets and MVO-ANN for classification purpose. The training method considered the capabilities of the MVO with respect to high exploration and exploitation to determine the optimal values of weights and biases of ANN-MLPs. This model applied to the problem of intrusion detection and is validated using the well-known dataset, NSL-KDD. The proposed method selected 22 features out of 41 to be the most important features that contribute much to enhance the performance of anomaly-based intrusion detection system. To indicate the value of feature selection and its impact on system performance, the accuracy, detection rate and false alarm rate are measured via two experiments with feature selection (MCF & MVO-ANN) and without feature selection (MVO-ANN) in a similar condition. The results reveals that the proposed method outperforms (MVO-ANN) as it demonstrates better and more robust results in terms of the stated measures and execution time.

Moreover, this model is compared existing models in a similar experimental setting. In terms of detection rate, false alarm rate and execution time, the proposed MCF&MVO-ANN outperforms the existing models, and results indicate that the proposed model could be effective for IDS. Further work will focus on improving the accuracy of classification to detect the type of attacks and hence the IDS model can be extended for multi-class classification problems. In addition, we will perform further tests on KDDCup99 dataset to validate the proposed model for the future researc.

REFERENCES

[1] 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.
[2] D. Papamartzivanos, F. Gomez Marmol, and G. Kambourakis, "Introducing deep learning self-adaptive misuse network intrusion detection systems," *IEEE Access*, vol. 7, pp. 13546–13560, 2019.
[3] M. Shehab, A. T. Khader, and M. A. Al-Betar, "A survey on applications and variants of the cuckoo search algorithm," *Appl. Soft Comput.*, vol. 61, pp. 1041–1059, Dec. 2017.
[4] V. R. Balasaraswathi, M. Sugumaran, and Y. Hamid, "Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms," *J. Commun. Inf. Netw.*, vol. 2, no. 4, pp. 107–119, Dec. 2017.
[5] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, Feb. 2019.
[6] Z. Cheng, J. Wang, M. Zhang, H. Song, T. Chang, Y. Bi, and K. Sun, "Improvement and application of adaptive hybrid cuckoo search algorithm," *IEEE Access*, vol. 7, pp. 145489–145515, 2019.
[7] J. H. Mohamud and O. N. Gerek, "Poverty level characterization via feature selection and machine learning," in *Proc. 27th Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2019, pp. 1–4.
[8] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO Metaheuristic," *Comput. Netw.*, vol. 168, Feb. 2020, Art. no. 107042.
[9] N. Singh and A. Kaur, "Feature selection for artificial neural network based intrusion detection system," *Int. J. Technol. Res. Eng.*, vol. 2, no. 11, pp. 2681–2683, 2015.
[10] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 420–432, May 2016.
[11] D. I. Mahmood and S. M. Hameed, "A feature selection model based on genetic algorithm for intrusion detection," *Iraqi J. Sci.*, pp. 168–175, Apr. 2018.
[12] J. Lee, D. Park, and C. Lee, "Feature selection algorithm for intrusions detection system using sequential forward search and random forest classifier," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 10, pp. 5132–5148, 2017.
[13] R. F. Najeeb and B. N. Dhannoon, "A feature selection approach using binary firefly algorithm for network intrusion detection system," *ARPN J. Eng. Appl. Sci.*, vol. 13, no. 6, pp. 2347–2352, 2018.
[14] E. M. Chakir, M. Moughit, and Y. I. Khamlichi, "An effective intrusion detection model based on SVM with feature selection and parameters optimization," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 12, pp. 3873–3885, 2018.
[15] G. Ke and Y. H. Hong, "The research of network intrusion detection technology based on genetic algorithm and BP neural network," *Appl. Mech. Mater.*, vols. 599–601, pp. 726–730, Aug. 2014.
[16] C. Qiu and J. Shan, "Research on intrusion detection algorithm based on BP neural network," *Int. J. Secur. Its Appl.*, vol. 9, no. 4, pp. 247–258, Apr. 2015.
[17] C. Lu, L. Zhai, T. Liu, and N. Li, "Network intrusion detection based on neural networks and D-S evidence," in *Image and Video Technology* (Lecture Notes in Computer Science), vol. 9555. 2016, pp. 332–343.
[18] I. Benmessahel, K. Xie, and M. Chellal, "A new evolutionary neural networks based on intrusion detection systems using multiverse optimization," *Int. J. Speech Technol.*, vol. 48, no. 8, pp. 2315–2327, Aug. 2018.
[19] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Comput.*, vol. 22, no. 1, pp. 1–15, Jan. 2018.
[20] I. Benmessahel, K. Xie, M. Chellal, and T. Semong, "A new evolutionary neural networks based on intrusion detection systems using locust swarm optimization," *Evol. Intell.*, vol. 12, no. 2, pp. 131–146, Jun. 2019.

[21] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.
[22] H. Faris, I. Aljarah, and S. Mirjalili, "Training feedforward neural networks using multi-verse optimizer for binary classification problems," *Int. J. Speech Technol.*, vol. 45, no. 2, pp. 322–332, Sep. 2016.
[23] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biologically Inspired Comput. (NaBIC)*, Dec. 2009.
[24] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, "Cuckoo search Optimization—A review," *Mater. Today, Proc.*, vol. 4, no. 8, pp. 7262–7269, 2017.
[25] P. Kora, "Crossover operators in genetic algorithms?: A review crossover operators in genetic algorithms?: A review," Tech. Rep., Mar. 2017.
[26] J. Singh and R. Banerjee, "A study on single and multi-layer perceptron neural network," in *Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2019, pp. 35–40.
[27] C. Chio, "Machine learning based techniques for network intrusion detection," in *Proc. Hack Paris*, 2016, pp. 79–83.
[28] A. G. M. Tavallaee, E. Bagheri, and W. Lu. *Canadian Institute for Cybersecurity*. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html
[29] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy and survey of intrusion detection system design techniques, network threats and datasets," vol. 1, no. 1, 2018.
[30] D. Houcque. *MATLAB for Engineering*. [Online]. Available: https://www.educba.com/introduction-to-matlab/
[31] A.-C. Enache and V. Sgarciu, "Anomaly intrusions detection based on support vector machines with an improved bat algorithm," in *Proc. 20th Int. Conf. Control Syst. Comput. Sci.*, May 2015, pp. 317–321.

SAMIRA SARVARI received the degree (Hons.) in Software Engineering from IAU-Iran and the master's degree in distributed computing from University Putra Malaysia (UPM), where she is currently pursuing the Ph.D. degree with the Computer Science Department, Faculty of Computer Science and Information Technology. Her research fields of expertise are computer and information security, machine learning, wireless local area networks, and recommendation systems.

NOR FAZLIDA MOHD SANI received the B.S.K. degree and the M.Sc. in computer science degree from UPM, and the Ph.D. degree UKM. She is currently an Associate Professor with the Faculty of Computer Science and Information Technology. Her research fields of expertise are information security, secure coding, authentication systems, program understanding, and debugging.

ZURINA MOHD HANAPI received the B.Comp.Sc. degree from Strathclyde, the M.Sc. degree from UPM, and the Ph.D. degree from UKM. She is currently an Associate Professor with the Faculty of Computer Science and Information Technology. Her research fields of expertise are computer system engineering, network security, and distributed computing.

MOHD TAUFIK ABDULLAH received the D.S.K., B.S.K., M.Sc., and Ph.D. degrees from UPM. He is currently an Associate Professor with the Faculty of Computer Science and Information Technology. His research fields of expertise are computer security and computer forensics.