A New Method for Intrusion Detection in Computer Networks using Computational Intelligence Algorithms

Yanrong HAO, Shaohui YAN*
Department of Software Engineering, Hebei Software Institute
Baoding 071000, China

***Abstract*—This paper introduces a novel and integrated approach to intrusion detection in computer networks that makes use of the benefits of both abuse detection and anomaly detection techniques. The proposed method combines anomaly detection and abuse detection technologies to enhance intrusion detection functionality. The intrusion detection system is implemented using a set of algorithms and models in the proposed approach. The frog jump algorithm has been utilized to choose the system's ideal input attributes. The decision tree is utilized in this system's abuse detection portion. Support vector machines or basic-radial neural network models have been utilized to find anomalies in this system. In the process of training neural networks, other techniques like particle swarm or genetic optimization are also utilized. The NSL-KDD dataset was used the experiment, and the findings were published. These findings demonstrate that, in comparison to using only anomaly or abuse detection, the proposed approach can increase the effectiveness of intrusion detection in the network. Additionally, a model that uses the frog leap algorithm for feature selection and classification and combines decision tree and support vector machine techniques with ten chosen input features has a detection rate of 98.2%. This is true despite the fact that the detection rates of the systems trained using comparable data in prior studies with 33 and 14 selected input features to the trainer have been 83.2% and 84.2%, respectively. Additionally, the algorithm execution performance increases up to 29 times faster than the aforementioned approaches when the intrusion detection rate is maintained at the level of other competing methods that were simulated in this work.**

Keywords*—Decision tree; network intrusion detection; particle swarm algorithm; basic-radial neural network; frog jump algorithm

I. INTRODUCTION

Computers connected to the Internet are threatened by a variety of things, including unauthorized access to user systems and the execution of unpleasant behaviors [1]. Typically, network penetration is viewed as an attack [2]. Intrusion detection systems are already a commonplace component of the security architecture. The following is a list of the intrusion detection system's objectives: Preventing behavioral issues that attack or abuse the system, recognizing attacks and coping with them, documenting current attacks, quality control, and giving security managers meaningful penetration information are just a few of the objectives [3].

According to the "CSI/FBI Computer Security and Crimes Survey" report, intrusion detection system usage increased from 42% in 1999 to 62% in 2010 and will reach 62% by 2022 has retained. These numbers demonstrate the critical role that these systems play in security technologies [4].

Network intrusion detection systems monitor network activity to find assaults. Exploit detection and anomaly detection are the two primary techniques for intrusion detection [5]. Using patterns and signatures that signal assaults, you can find exploits and intrusions. Detects assaults but is unable to identify them [6]. Due to the use of less complex identification algorithms, the abuse detection technique has the advantage of extremely rapid identification [7]. Activities that depart from regular functioning are identified as infiltration in the anomaly detection approach by the construction of normal usage profiles. Because of this, the exploit detection approach is unable to identify unexpected intrusions that the anomaly detection system can [8]. The high prevalence of false alarms in the anomaly detection approach is one of its shortcomings [9].

Intrusion detection systems that integrate both strategies have been developed to address the issues with these two approaches [10]. These systems do it in three different ways. The following approaches are used: a. abnormality is first identified, followed by abuse; b. parallel approach; and c. abuse is first identified, then abnormality [11].

The detection rate of all sorts of assaults (known and unknown) increases because under the parallel approach, incoming traffic is evaluated independently by each method (identification of abuse and identification of anomalies) [12]. The anomaly detection approach still has a high risk of false positive notifications, but if the detection model qualifies the communication as an attack, this method likewise treats the incoming communication as an assault [13, 42]. The computational cost of detection is another concern, as each communication must be examined using both anomaly detection and abuse detection models, which raises this cost [14].

In the combined strategy employed in this article, we have attempted to produce an ideal plan by combining already-existing algorithms and models [15]. In this regard, it has been managed to minimize the number of input features to the system from 41 to 10 thanks to the deployment of the optimization technique based on frog jump. Comparing this

strategy to several studies in this subject, the combined system is also one of the drawbacks. In this manner, the abuse detection system receives the inbound traffic first (Fig. 1). The exploit detection phase's outputs that do not fit the intrusion patterns are fed into the anomaly detection system as input in order to find unidentified intrusions [16, 43]. The effectiveness of anomaly detection declines as the volume of attacks rises. Hence exploit detection has been used first to address this issue. Known attacks can be found using exploit detection. The quantity of attacks required to find anomalies is drastically decreased by eliminating known attacks. Another benefit is that the suggested hybrid system can identify known intrusions in real-time due to the high speed of exploit detection techniques such as decision tree algorithms [17].

In this situation, the plan's anomaly detection component uses well-established, successful models (such as artificial neural networks) that can recognize novel attacks [6]. Instead of employing conventional methods for neural network training, computational intelligence algorithms have been adopted since they have proven to be more effective [18].

The main motivation of this research is to improve the effectiveness of intrusion detection systems (IDS). By combining abuse detection and anomaly detection methods, this approach aims to increase the system's ability to detect and mitigate various types of network intrusions. Traditional IDSs may struggle to effectively identify both known and unknown threats. Another motivation is to optimize the use of computing resources and reduce processing time. The choice of algorithms such as frog jumping algorithm, decision trees and support vector machines (SVM) shows the desire to achieve efficient and accurate intrusion detection without computational overhead. In summary, the proposed approach in this research is motivated by the need for more effective intrusion detection systems; more efficient and adaptable in the face of evolving cyber threats. Potential benefits include improved detection rates, resource optimization, reduced false positives, and increased consistency, all of which contribute to a stronger and more comprehensive network security solution. In this regard, the suggested method's usage of radial basis neural network (RBF) in the anomaly detection phase has also led to a notable increase in the algorithm's execution speed when compared to other hybrid techniques. A comparison of the proposed hybrid system's performance with other similar schemes that have been tried on the same field with related events reveals that the suggested scheme has attained the desired detection rate (see Table VII in Section V). The main contributions of this research and their possible consequences are as follows:

- Hybrid intrusion detection system: This research introduces a new hybrid intrusion detection system that combines abuse detection and anomaly detection techniques. This hybrid approach can lead to more effective and robust intrusion detection because of its strengths. Both methods are used. The result is improved network security and a greater likelihood of detecting a wide range of intrusions.
- Feature selection and data preprocessing: This research emphasizes the importance of feature selection and data preprocessing techniques to improve the quality of input data. Choosing the right feature and normalizing the data can lead to a more reliable and accurate intrusion detection system. The consequences are higher accuracy in detecting intrusions and reduction of noise in the data.

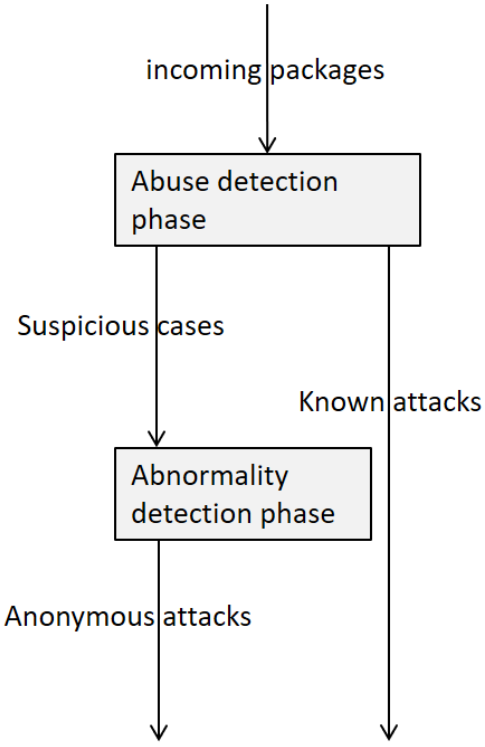


Fig. 1. Hybrid intrusion detection system with abuse detection priority.

- Efficiency and scalability: This research deals with the efficiency and scalability of the intrusion detection system, especially by comparing SVM and radial basis neural networks. Understanding the trade-offs between different algorithms and their impact on performance and scalability can help deploy intrusion detection systems in different network environments. The result is consistent and resource-efficient intrusion detection.

The remainder of the essay is structured as follows. Studying earlier works is the subject of Section II. Section III offers suggestions for spotting abuse and abnormalities. Section IV discusses evaluation and simulation results, and Section V concludes with recommendations for future research.

II. Background Research

This section will first cover some earlier research on hybrid systems before introducing the fundamental models and techniques employed in this study.

A. Related Work

Some of the studies on hybrid systems have been covered in this subsection. The three-layer architecture of the intrusion detection system, which was created and developed by [19], is an illustration of a hybrid system. Based on the KDD Cup'99 standard data, the system featured a blocklist, an allowlist, and a multiclass support vector machine bundle. Blocklists are used to filter out known threats from network traffic, whereas allowlists are used to identify legitimate traffic.

A method for audit data mining and analysis (ADAM) was presented by [20] in which abuse detection is used after anomaly detection. In order to identify attacks, ADAM combined an association mining rule with a classification mechanism. The suspect traffic is first identified by the association mining rule-based anomaly detection model, which then sends it to the abuse detection model. After then, the suspicious communications are classified by the abuse detection model as "normal," "known attacks," and "unknown attacks" (false alert of the anomaly detection model). Its usage is uncommon. Communications that cannot be categorized as typical patterns or known attacks under the ADAM technique are categorized as unknown attacks. If anomaly detection is used first, then abuse detection, the anomaly detection model should have a high detection rate, and the abuse detection model should eliminate the false alarms generated by the anomaly detection model by differentiating between known and unknown attacks. The majority of abuse detection systems, however, are ineffective at lowering false alerts. An anomaly detection model, an abuse detection model, and a decision support system were all incorporated in the [21] proposal for an intelligent hybrid intrusion detection system. They used a decision tree to model the abuse detection model and a self-organizing map (SOM) neural network to model anomaly detection. Following independent training of each model, the decision support system pooled the categorization outcomes of the two models.

A hybrid intrusion detection system was designed in [22] using the abuse detection approach first, then the anomaly detection method. The exploit detection model is quicker than the anomaly detection model and can identify known attacks with a low probability of false positives. In order to identify known attacks, the Al-Teda detection and exploitation model was employed. Only non-deterministic connections were then detected using the anomaly detection model. A technique for detecting anomalies identifies outliers that deviate from typical data patterns and classifies them as well-known assaults. The anomaly detection and abuse detection models, however, are trained independently, just like the parallel hybrid technique, which causes a high risk of false positive notifications in the outcomes.

With a hybrid methodology, researchers in [23] initially used the C4.5 decision tree algorithm to evaluate the traffic during the abuse detection phase. After the exploit detection phase, in the anomaly detection phase, distinct support vector machines (SVM) were employed to discover unexpected incursions for each subset of data classified as normal by the intrusion detection model. Each subset will be more effective at generating typical profiles and finding anomalies because it has more concentrated data.

Computational intelligence techniques have been utilized for feature selection and decision trees for classification in numerous studies in the area of computer network security. For instance, researchers in [24] utilized the decision tree with the C4.5 training method to identify garbage items and the binary version of the particle swarm optimization technique (BPSO) for feature selection. The ideal collection of features was likewise chosen by the source [25] using CFA-based optimization, and the chosen features were assessed using a decision tree-based classifier. The estimation and selection of acceptable and chosen features for data classification during the tree training process is one of the decision tree's beneficial properties. On the other hand, adding a feature selection phase before training the decision tree has been shown in experiments to significantly improve the classification accuracy of the decision tree [26]. As a result, the frog jump method incorporated a feature selection step before training the decision tree in the article's suggested solution. The techniques and models used in this article are briefly introduced in the sections that follow so that the next sections can explore how to combine them and express the simulation results.

B. Frog's Leap Algorithm

The combined optimization method based on frog jumping (SFLO) is one of many evolutionary algorithms that have been created in recent decades to decrease processing time and increase the quality of results [24]. This program uses a technique called imitative discovery [27] and aims to use a heuristic search to identify the overall best answer. In order to discover effective general solutions, this technique has been tested on a number of combinatorial problems. The population of potential solutions for the frog leaping algorithm is defined as a set of subsets of frogs (solutions). Distinctive subgroups are viewed as distinct frog species, each of which conducts a distinctive local search. Each frog in the subgroups has ideas that are shaped by those of other frogs and changed by the process of imitational evolution. Ideas spread through the subsets through the process of interweaving and interweaving after going through the evolutionary phases of imitation. Until

the convergence rule is satisfied, local search and nesting operations are carried out [28].

C. Decision Tree

One of the most well-known techniques for creating a classification model is the decision tree. The result information of decision tree-based classification algorithms is displayed as a tree of several feature value states. Always dividing records based on a candidate feature that maximizes a particular criterion is a greedy approach to decision tree construction. The most deserving feature will be the one that improves the tree the most in accordance with this criterion. Depending on how the features of the data set are chosen to be included in the decision tree and when to cease growing the tree, there are various types of decision trees. Better than other failures are one where the distribution of bundles in the resulting nodes is homogeneous. The node is homogeneous if all of its records are suspended in the same category. Since, in this situation, that node turns into a leaf. In actuality, the node with the least number of impurities is the homogeneous node. They are placed in the interest relationship, and the amount of interest resulting from each failure is calculated after the amount of impurity arising from each failure has been determined. The failure-related impurity is removed from the parent node's impurity in the gain relation. Any failure that generates a greater profit is preferable, and that failure will ultimately be chosen [26]. The C4.5 tree, which is the decision tree employed in this article, is utilized to determine its impurity using the entropy approach based on Eq. (1):

$$\text{Entropy}(t) = -\sum_i p(j|t)\,\text{logp}(j|t) \quad (1)$$

In relation (1), p(j|t) denotes the proportion of records belonging to the jth category to all other records in node t. The fracture gain is calculated after the entropy for each node has been determined, followed by the entropy for the entire fracture. A failure is better if it has a greater interest rate. The gain of a failure is calculated using Eq. (2):

$$\text{GAIN}_{\text{split}} = \text{Entropy}(p) - \sum_{i=1}^{k} \frac{n_i}{n} \text{Entropy}(i) \quad (2)$$

Regarding this, n represents the overall number of records in the parent node, n_i represents the number of records in the i_{th} child, entropy(p) represents the entropy of the parent node, and entropy(i) also represents the entropy of the i_{th} node [29]. The test data set can then be used to test the classification model that was created using the decision tree. The goal of using the model is to predict, using the model, the category attribute value for the test record.

D. Basic-Radial Neural Network (RBF)

The artificial neural network has the benefit of generalization, which sets it apart from other classifiers. With a small amount of training data, radial basis function networks are frequently used to estimate multidimensional functions nonparametrically. The RBF network is highly helpful since it trains quickly and thoroughly. With enough neurons in the hidden layer, it can estimate any continuous function with any level of accuracy. Fig. 2 depicts the three-layer network that makes up RBF's main design.

The basic-radial functions are shared by the neurons in the intermediate (hidden) layer [30]. According to Eq. (3), the third layer approximates the middle layer neurons' output by summing their weighted output:

$$F(x) = \sum_{i=1}^{p} w_j \emptyset(||x - u_j||) \quad (3)$$

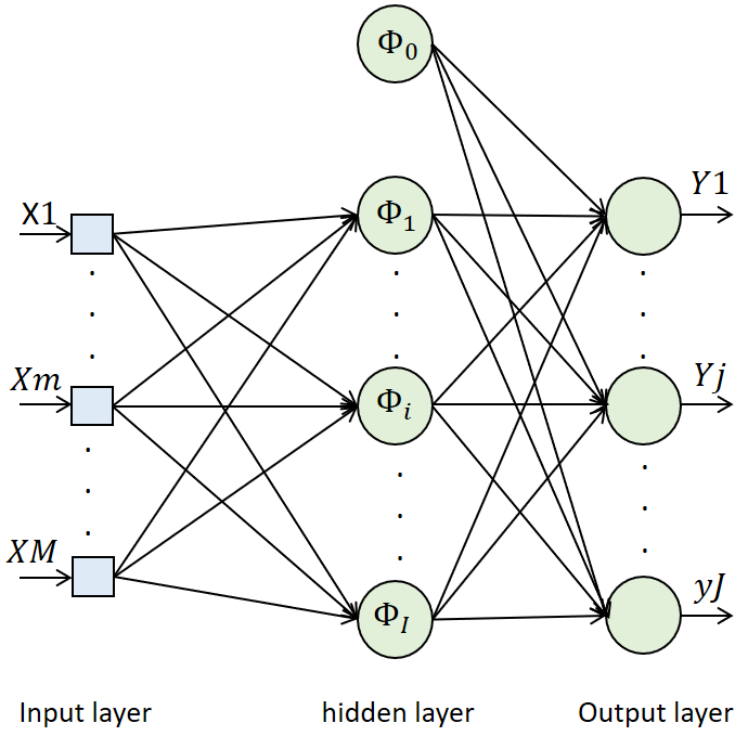


Fig. 2. The architecture of an RBF network [30].

If RBF is used to approximate the function, its output will be helpful. However, a hard limiter can be applied to the output neurons to ensure output values of 0 or 1 for pattern classification purposes. The base-radial function p with centers u_j is utilized for the approximation of the F function based on Eq. (3). Sign ‖.‖ the exponent, which is typically set to represent the Euclidean distance, is the distance function of R_n space. The most well-known basic-radial function, which has been proposed, is the Gaussian function in RBF networks. The Gaussian function, an exponential function from the category of functions with the best approximation qualities, was chosen as the response function of neurons in RBF networks. This ensures that there is a set of weights that more accurately approximates the relationship between the input and target vectors than any other set. The sigmoid function utilized in the creation of error backpropagation networks makes use of this assurance that it has no existence.

E. Particle Swarm Optimization Algorithm

The social behavior of a group of birds is described by the population-based stochastic optimization algorithm known as the PSO algorithm. In space, a flock of birds looks for food randomly. Following the bird that is closest to the food can be one of the greatest tactics. The PSO algorithm's core concept is this tactic [31]. The search space for the PSO algorithm is analogous to the search space for the bird movement pattern. In the PSO algorithm, each solution, also known as a particle, is analogous to a bird, and there are exactly as many particles (solutions) as there are birds. Each particle has a merit value, which is determined by a merit function. The more merit a particle has, the closer it is to the target in the search space, which in the bird movement model is food. Additionally, every particle has a displacement that controls its direction of motion and is used to predict its next location. Until it ultimately reaches the ideal solution, each particle moves forward in the search space by adhering to the best particles in the current state [32]. The particle velocity vector (Vi) and particle (Xi) in the (t+1)th iteration are computed from relations (4) and (5) in each step of the particle swarm algorithm iteration:

$$V_i(t+1) = w \times V_i(t) + C_1 \times \text{rand}_1 \times [\text{pbest}_i(t) - X_i(t)] + C_2 \times \text{rand}_2 \times [\text{gbest}_i(t) - X_i(t)] \quad (4)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (5)$$

F. Hereditary Algorithm

Hand first proposed the genetic algorithm in 1965. From among the chromosomes in a population, the selection operator chooses the number of chromosomes for reproduction. Fitter chromosomes are more likely to be chosen for reproduction. Chromosome segments are randomly switched between during crossing over. Because of this, the children don't exactly resemble one of their parents but instead exhibit traits from both. A single-point, two-point, or even intersection could exist [33].

All chromosomal points have an identical chance of merging during the uniform crossover. In this case, any valley can be used to pick the child's chromosome genes. It is possible to perform uniform intersection using relations (6) and (7):

$$y_{1i} = a_i x_{1i} + (1 - a_i) x_{2i} \quad (6)$$

$$y_{2i} = a_i x_{2i} + (1 - a_i) x_{1i} \quad (7)$$

In these relationships, x_1 and x_2 are the parents, while y_1 and y_2 are the first and second children, respectively. Additionally, the values of a_i in continuous issues are in the range [0,1], while those in binary problems are equal to zero or one. The letter i stands for the input dimensions (solution space dimensions).

The chromosomes are subjected to the mutation operator after crossing over. This operator chooses a gene at random from a chromosome and modifies the information within that gene. If the gene is a binary number, it is inverted; if it is a member of a set, another value or component of that set is substituted for the gene. The created chromosomes are known as the new generation and are sent to the following round of algorithm execution after the mutation operation is finished.

III. Suggested Method

Fig. 1 shows the suggested method's operating principles. The specifics of spotting abuse and abnormalities will be covered in this section. Prior to applying the data to the model in the proposed combined method, the data was first preprocessed using the leapfrog computational intelligence algorithm to extract its key features, which improved the system's overall performance and, in particular, the effectiveness of the decision tree. The regular training data is then separated into subgroups during the misuse detection stage whose connection patterns have less variation than the entire normal data. Then, in the anomaly detection stage, a different anomaly detection model is applied for each subset. As a result, each subset's efficiency will be higher in producing more normal profiles and rejecting the result in anomaly detection since each subset has more concentrated data [9, 34]. The quality of the input data to the anomaly detection stage is not good enough and has a significant impact on the accuracy of the SVM network if the decision tree is unable to appropriately partition the data into subsets while retraining the model in the operational environment. This lowers the effectiveness of the SVM network; however, the RBF neural network is not constrained by this issue. Since model training is a linear process and neural network training is slower, the effectiveness of the models is unaffected [34].

The RBF neural network was employed in the anomaly detection phase, and the effectiveness of SVM and RBF was compared in the anomaly detection phase, in accordance with the foregoing and to prevent the reduction of the accuracy of the SVM network, in cases where the output of the decision tree is not effective. It is important to note that switching from SVM training to RBF training and performing feature selection prior to the abuse detection stage in order to condense the problem's dimensions were the adjustments that significantly increased the effectiveness of the newly proposed method. The simulations were conducted using Matlab software version 2022a as well. The steps for intrusion detection in the suggested approach are as follows:

- Data preprocessing includes the following steps:

a) Data homogenization (training and test data) by replacing the letter characters of the data set with numerical values

b) Data normalization (training and test data): In this step, the values of the continuous features are normalized to the interval [-1,1] according to Eq. (8).

$$X = 2 \times \frac{x - \min(x)}{\max(x) - \min(x)} - 1 \quad (8)$$

c) Reducing the dimensions of the problem by feature selection by the leapfrog computational intelligence algorithm

It should be noted that normalization (or rescaling of features) removes the imbalance between the data [36-34]. In the dataset used in this paper (NSL-KDD), some features have large numerical values that can dominate other features. For example, the dst_bytes attribute can have values from zero to about 1.3×10^9, while the same_srv_rate attribute has values between zero and one [35].

- Applying various methods in the combined model of penetration detection:

(First the abuse detection phase and then the abnormality detection phase) including the C4.5 decision tree in order to detect abuse and then apply separate RBFs in the abnormality detection phase for the categories that are detected as normal. It is reminded that the training parameters in the RBF neural network are determined with the help of the genetic algorithm or the particle swarm algorithm. The block diagram presented in Fig. 3 shows the working steps in the proposed intrusion detection method in more detail.

A. The Stage of Identifying Abuse

In the data mining process, the data are ready to be applied to the model learning stage after feature selection and data pretreatment. The C4.5 decision tree is utilized for this. The data exploration strategy that is chosen will determine the sequence guiding the preprocessed data during the learning stage, and the created model will then be sent to the evaluation stage (i.e., evaluation and interpretation of the model) for evaluation. It was time to cut the decision tree after training and preserving it, leaving only 12 leaves with conventional labels. Too small sets make this operation excessively slow because it takes time to split training data into distinct subsets based on various rules. The tree was pruned for these reasons because, on the other hand, the restriction and lack of overdispersion of neural network inputs during the training phase will reduce the generalization capacity of the network.

B. Anomaly Identification Stage

At this point, distinct RBFs were utilized for each of the remaining leaves with the conventional label, and the data input for each of them was identical to the information classified by the decision-making process. The neural network's leaves had come. The conditions of their development were determined for all the leaves with the normal label in order to obtain the data in each leaf. Based on these conditions, the training data set was then segmented into several input subsets, and for each of them, different neural networks were employed. The usage of a more homogeneous data set boosts the network's accuracy, and the anomaly detection system is better prepared to deal with anomalies that do not follow the typical pattern since it is more accustomed to normal patterns with lower dispersion [36].

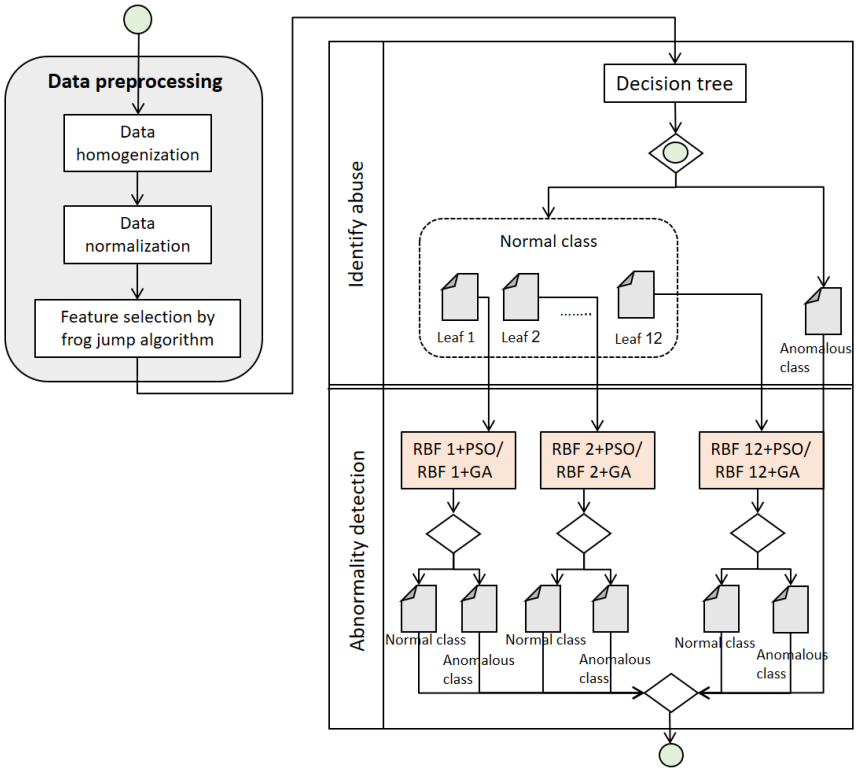


Fig. 3. Process block diagram of the proposed intrusion detection method.

IV. Evaluation Results of the Proposed Solution

A. NSL-KDD Dataset

The NSL-KDD dataset has been utilized to evaluate and contrast the performance of the suggested method [37]. The NSL-KDD dataset is an edited version of the KDD'99 dataset that was made available as a result of KDD'99's issues with the presence of numerous duplicate samples in the training and test data, as well as the resolution of these issues. The 41 characteristics in this dataset span a wide range of numerical values and are of continuous, discrete, and symbolic types. As previously stated, data preprocessing is required to correct the imbalance in the entire dataset and eliminate the impact of scale differences in such a database [38]. In other words, normalization in such data on all data (in the following, some of them as training data and others as training data) tests are selected) [39] in order to prevent features with high numerical values from overpowering other features.

B. Overall Evaluation Results

The frog jump algorithm was used to intelligently choose ten attributes from among them in order to decrease the problem's dimension. Feature selection (variable reduction) aids in data comprehension, lower processing demands, lessens the impact of the dimensionality problem and enhances prediction performance. The goal of feature selection is to choose a subset of input variables that can accurately characterize the input data while minimizing the impact of extraneous factors and producing accurate prediction results [40].

In this step, the trained decision tree was first given all the input data from the abuse detection phase (signature detection) in order to evaluate the model. Assuming they are unidentified attacks whose signatures or data packet characteristics could not be recognized by the exploit detection model; the identification component will once more detect the data that the decision tree classified as regular packets. In this phase, anomalies (trained RBF) were examined to determine their class. The way the anomaly detection phase operates is that the inputs are first reviewed again in accordance with the requirements of the decision tree's leaves to choose which of the RBFs should be provided as input. The selective RBF then determines the data packet's final class.

The time needed to run the model is significantly reduced if RBF is used in place of SVM, according to the results, so that the time needed for the test is decreased by an average of 26 times (if PSO is employed), a 24 times improvement, and if we employ GA, we will see an improvement in execution time of more than 28 times); however, the model error had a modest decline.

It should be noted that the ideal number of kernels for each RBF was found independently during the RBF training procedure. If PSO and inheritance algorithms are applied, the number of optimal cores for every RBF may be deduced from Fig. 4 and 5, respectively. According to the number of various kernels, each line in the graphs depicts the MSE error trend for one of the RBFs, and the number of optimal kernels for each RBF is the same as the number of kernels that minimize the MSE error.

As previously indicated, PSO and genetic algorithms (GA) were used to train the RBF neural network individually, and the outcomes of both were compared. The results were nearly identical, and no discernible difference was found regarding the effectiveness of applying PSO or GA in RBF training.

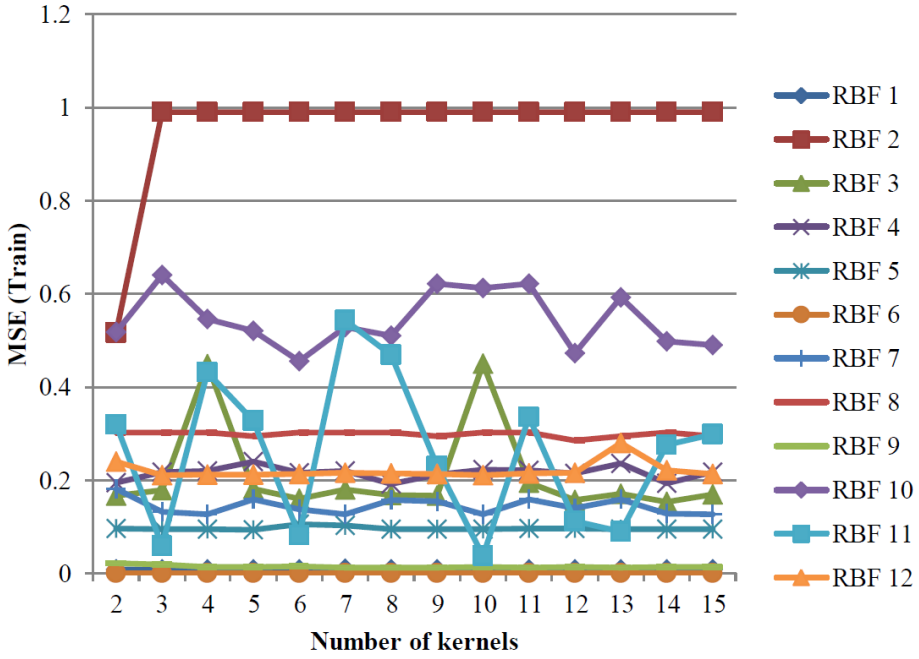


Fig. 4. Training error of all RBFs using PSO algorithm for different numbers of kernels.

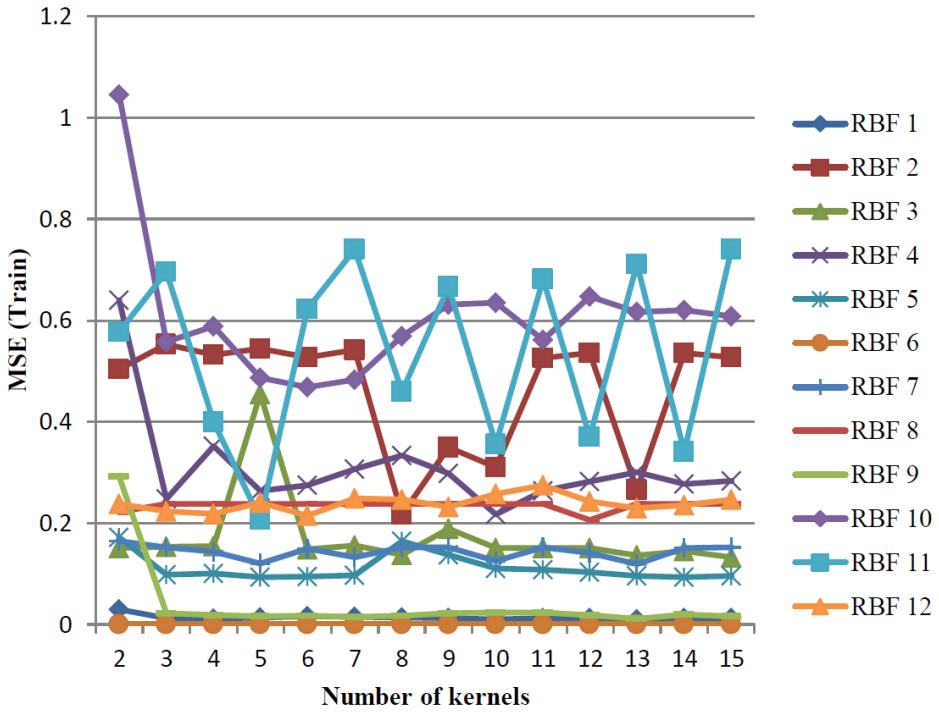


Fig. 5. Training error of all RBFs using GA for different numbers of kernels.

In accordance with relations (9) to (12), the following metrics were employed to assess the models: mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and set of square error (SSE).

$$MSE = \sum_{i=1}^{n} \frac{(t_i - y_i)^2}{n} \quad (9)$$

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(t_i - y_i)^2}{n}} \quad (10)$$

$$MAE = \sum_{i=1}^{n} \frac{|t_i - y_i|}{n} \quad (11)$$

$$SSE = \sum_{i=1}^{n} (t_i - y_i)^2 \quad (12)$$

In the relationships mentioned above, n denotes the number of samples, t_i is the goal output value, and u_i denotes the output value of the class that the model predicted [41]. An Intel Core i7 machine with a clock speed of 2.1 GHz and 6 GB of random-access memory was used for the models' training and testing phases. Sixty-two thousand eight hundred sixty-four data points were used to train the models, while 85347 data points were used to test the models. The outcomes of using the models are shown in the paragraphs that follow.

C. Decision Tree Training Results

Table I shows the results of errors in detecting abuse. The time required to train the tree was equal to 0.18 seconds.

D. The Results of the Combined Model of Decision Tree and RBF with Training by PSO Algorithm

In Table II, the results of using the combined intrusion detection model are presented in the condition that the RBF model is trained with the PSO algorithm. The errors presented in Table II were obtained for this setting of values for the PSO algorithm parameters: number of population members: 60, personal learning coefficient (C1): 1.4962, collective learning coefficient (C2): 1.4962, W coefficient: 1 and Reduction factor W: 0.9.

E. The Results of the Combined Model of Decision Tree and RBF with Training by GA

Table III also shows the results of using the combined intrusion detection model in the conditions where the RBF model is trained with the genetic algorithm. The errors presented in Table III were obtained for this setting of values for the parameters of the genetic algorithm: number of population members: 20, crossover probability: 0.9, mutation probability: 0.3, and mutation rate: 0.6.

TABLE I. ERROR-VALUES OF TRAINING AND TESTING IN THE PHASE OF DETECTING ABUSE BY DECISION TREE

Phase	MAE	MSE	SSE
Education	0.00200	0.00401	252
Test	0.1017	0.2034	17360

TABLE II. TRAINING ERROR VALUES OF INDIVIDUAL RBFS – TRAINING BY PSO ALGORITHM

Model	MAE	RMSE	MSE	SSE
RBF 1	0.00376	0.08677	0.00753	196
RBF 2	0.25846	0.71897	0.51692	12636
RBF 3	0.07680	0.39190	0.15359	828
RBF 4	0.09631	0.43888	0.19262	4228
RBF 5	0.04696	0.30646	0.09392	760
RBF 6	0.00011	0.01499	0.00022	4
RBF 7	0.06345	0.35622	0.12690	2744
RBF 8	0.14286	0.53452	0.28571	136
RBF 9	0.00591	0.10872	0.01182	680
RBF 10	022767	0.67480	0.45535	928
RBF 11	0.01862	0.19299	0.03724	20
RBF 12	0.10501	0.45827	0.21001	76

TABLE III. TRAINING ERROR VALUES OF DISCRETE RBFS-TRAINING BY GA

Model	MAE	RMSE	MSE	SSE
RBF 1	0.00536	0.10350	0.01071	140
RBF 2	0.10873	0.46632	0.21745	2664
RBF 3	0.06645	0.36454	0.13289	360
RBF 4	0.10859	0.46604	0.21719	2388
RBF 5	0.04647	0.30485	0.09293	376
RBF 6	0.00023	0.02123	0.00045	4
RBF 7	0.34446	0.34446	0.11865	1276
RBF 8	0.10317	0.45426	0.20635	52
RBF 9	0.00598	0.10938	0.01196	344
RBF 10	0.23392	0.68399	0.46784	480
RBF 11	0.10370	0.45542	0.20741	56
RBF 12	0.10724	0.46312	0.21448	400

Additionally, Table IV provides the amount of time needed to train the simulated models. Because PSO and GA have different populations with different numbers of population members, RBF training and PSO and GA differ in innovative ways. Each model with the least population members that leads to the lowest error rate has been taught since increasing the number of population members in population-based optimization algorithms increases training time because more iterations are required for more population members. As can be seen, using SVM rather than RBF cuts down on training time during the anomaly detection stage. This is because RBF training involves using the aforementioned smart optimization algorithms to find the ideal network weights, and finding these ideal values requires a significant amount of repetition. However, because the training process takes place offline, it is acceptable to extend it in order to enhance the performance of the model's online execution.

F. The Results of Testing the Models with Test Data

The experimental results demonstrate that the model performs poorly when used in the anomaly detection phase with a significant amount of input data, which eventually results in the model's inefficiency in online applications (Table V). As can be shown, using a decision tree and an RBF neural network together (with training via the PSO algorithm) is the optimum option in terms of the model's online execution time. The aforementioned alternative offers competitive values and comes quite near to the combined decision tree and RBF neural network model (with GA training) in terms of the number of various error criteria (Table VI).

TABLE IV. THE TRAINING TIME OF THE MODELS

proposed model	Anomaly detection model training time (sec)	Time required to provide data for anomaly detection model (sec)	Training time of the misuse detection model (sec)	total time (sec)
C4.5+SVM	29.48314	17.79945	0.18482	47.4641
C4.5+FBF-PSO	78.76758	17.79945	0.18482	96.75185
C4.5+RBF-GA	59.21933	17.79945	0.18482	77.20360

TABLE V. TESTING TIME OF MODELS WITH TEST DATA

proposed model	total time (sec)
C4.5+SVM	77.216326
C4.5+RBF-PSO	2.701057
C4.5+RBF-GA	3.229697

TABLE VI. THE TESTING ERROR OF PROPOSED MODELS WITH TEST DATA

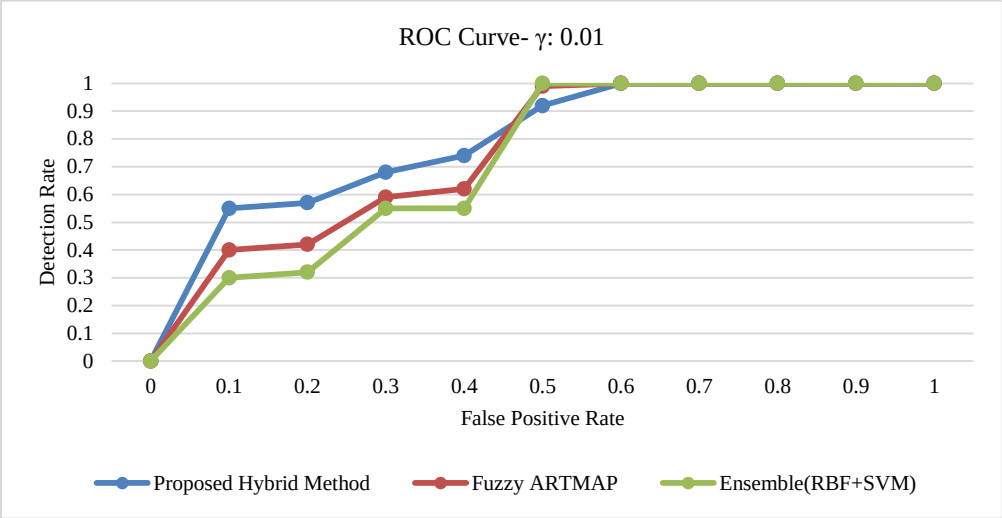
proposed model	MAE	RMSE	MSE	SSE
C4.5+SVM	0.17470	0.59110	0.3494	29820
C4.5+RBF-PSO	0.16685	0.57766	0.3337	28480
C4.5+RBF-GA	0.16645	0.57697	0.3329	28412

TABLE VII. COMPARISON OF THE DETECTION RATE OF THE PROPOSED SYSTEM WITH SIMILAR SYSTEMS

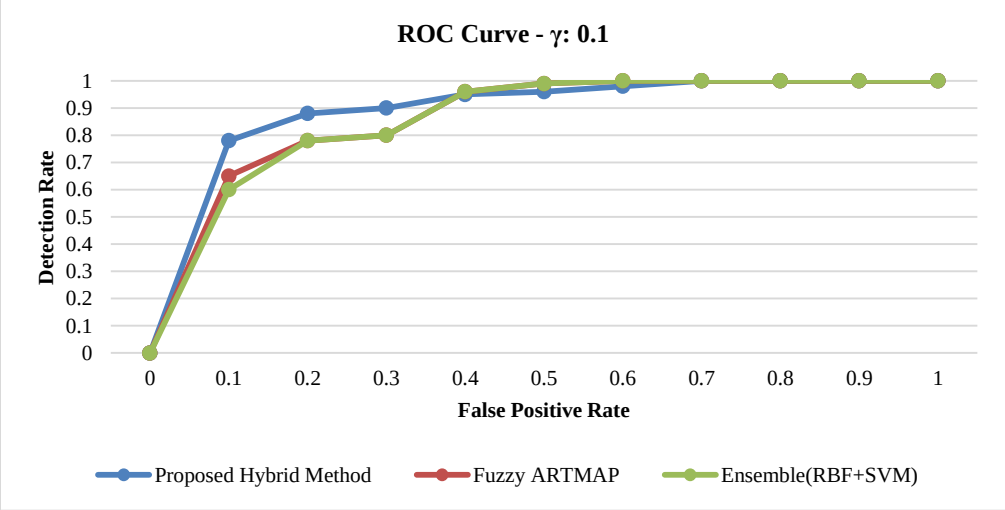
Abbreviated name of the Amikhtar model	Feature selection algorithm	Classification tool	Data Names	Number of selected features	Detection rate (%)
$\mathrm{FGBARM}^{1} + \mathrm{GSA}^{2} - \mathrm{Fuzzy\ ARTMAP}$[28]	FGBARM	GSA-Fuzzy ART MAP	KDD'99	26	93.74
FGBARM+PSO-Fuzzy ARTMAP[31]	FGBARM	PSO-Fuzzy ART MAP	KDD'99	29	97.25
FPGBARM+PSO-Fuzzy ARTMAP[32]	FGBARM	GA-Fuzzy ART MAP	KDD'99	29	9790
$\mathrm{DBN}^{3} + \mathrm{SVM}$[33]	DBN	SVM	NSL-KDD	14	83.14
$\mathrm{RST}^{4} + \mathrm{SVM}$[34]	RST	SVM	KDD'99	29	89.36
Info-Gain+J48[27]	Info-Gain	J48 decision tree	NSL-KDD	33	81.94
Info-Gain+Natve Bayes[36]	Info-Gain	Natve Bayes	NSL-KDD	33	75.79
$\mathrm{Info-Gain} + \mathrm{MLP}^{3}$[18]	Info-Gain	MLP	NSL-KDD	33	73.55
Info-Gain+SVM[41]	Info-Gain	SVM	NSL-KDD	33	71.02
$\mathrm{Info-Gain} + \mathrm{CART}^{6}$[42]	Info-Gain	CART	NSL-KDD	33	82.32
Ensemble+Bayesian[43]	Applying nine basic features, 13 content features, and 19 traffic features to three separate categories	Fuzzy K-NN, MLP, and Naive Bayes classifiers	KDD'99	A total of 41 features are used in three categories	93.35
Ensemble (RBF+SVM)[21]	Best First Search	RBF+SVM	NSL-KDD	9	85.19
SFLO+C4.5-SVM (recommended model)	SFLO	C4.5-SVM	NSL-KDD	10	97.4
SFLO+C4.5-PSO-RBF (recommended model)	SFLO	C4.5-PSO-RBF	NSL-KDD	10	96.9
SFLO+C4.5-GA-RBF (recommended model)	SFLO	C4.5-GA-RBF	NSL-KDD	10	96.9

With a false positive rate of 1.3%, DT has a detection rate of 99.2% for known threats and 31.06 percent for unknown attacks. DT turns out to be unsuitable for detecting new assaults but has strong detection performance for known attacks. The proposed method's detection performance was carefully examined in terms of anomaly detection since its goal is to enhance the detection performance of the anomaly detection model. Fig. 6 compares the suggested method's receiver operating characteristic (ROC) curves for unknown attacks with those for increases in parameter γ from 0.01 to 1. The proposed method has a greater detection rate than traditional ones. The rate of false positives in the decision boundaries of the anomaly detection model in the proposed method can depict normal behavior better than existing methods since each class 1 SVM model can concentrate on its corresponding decomposed region.

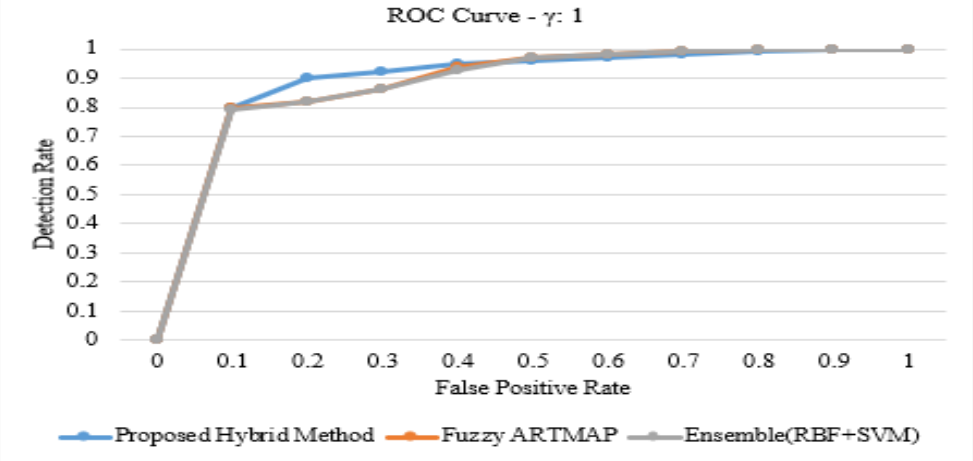
The proposed method's detection rate is roughly 11% greater than that of traditional approaches when the false positive rate is below 11%, which is ideal. It was found that the conventional method's detection rate improves when the false positive rate is about 51%. This outcome is believed to be the consequence of the suggested technique, which calls for building a class 1 SVM model for each deconstructed region. Instead of concentrating on each deconstructed zone when the false positive is very significant (like 51%), it is preferable to concentrate on the highly concentrated regions. It can be deduced that the detection performance of the suggested method is superior to traditional methods for unknown assaults because an IDS operator should have a very low false positive rate.



(a) Unknown assaults with parameter γ=0.01 on ROC Curve.



(b) Unknown assaults with parameter γ=0.1 on ROC Curve

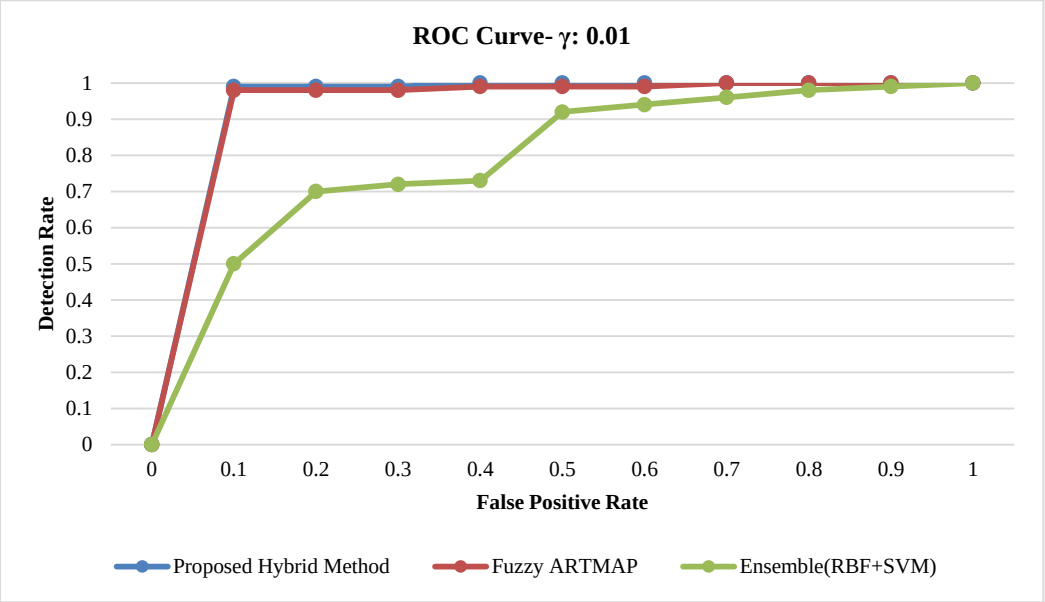


(c) Unknown assaults with parameter γ=1 on ROC Curve

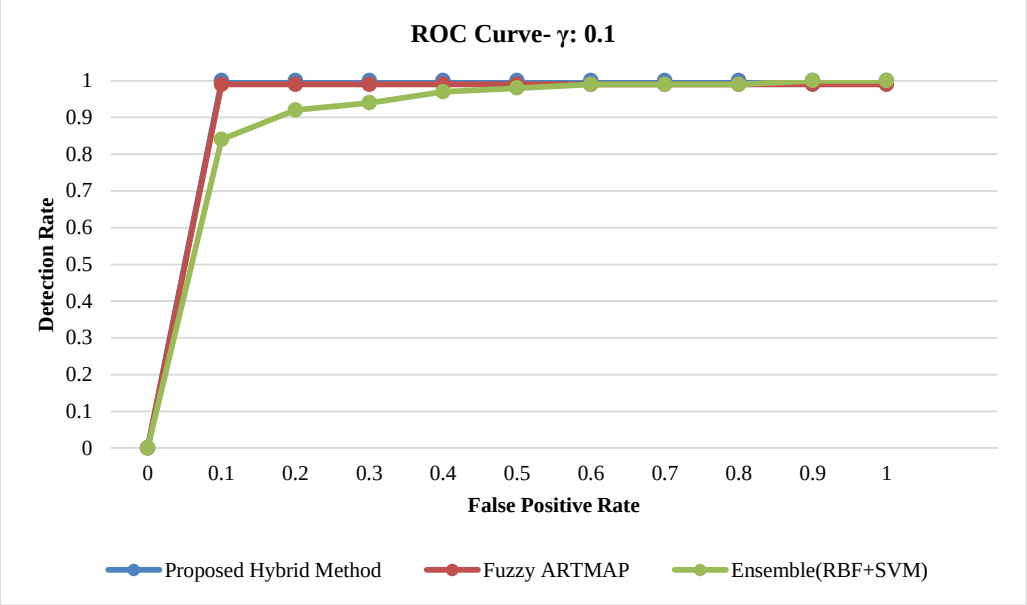
Fig. 6. Investigation and comparison of detection efficiency for unknown attacks using ROC curves.

G. The Class 1 SVM Model's Detection Rate Approaches that of the Traditional Hybrid Method as

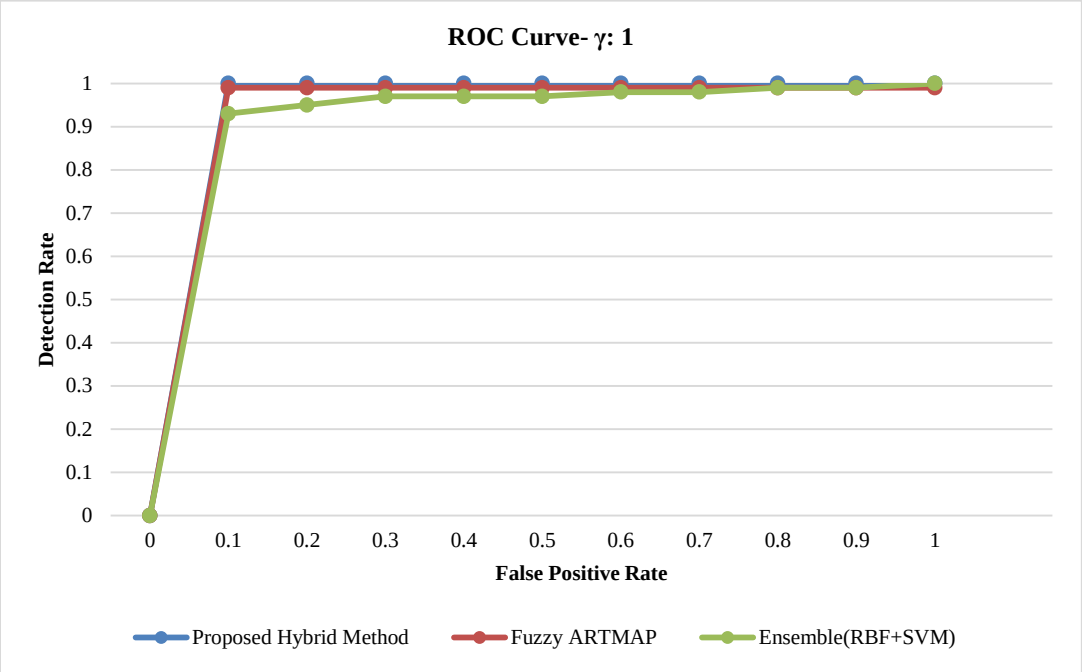
The parameter γ rises. This demonstrates that DT cannot influence the identification of unknown threats by enhancing Class 1 SVM performance. However, it has a major impact on hybrid intrusion detection models' ability to identify known attacks. Fig. 7, which varies the parameter c from 0.01 to 1, displays the ROC curves of the known attacks for the proposed technique and its comparisons. The figure shows that regardless of performance, both combined detection approaches from the SVM class 1 model have a detection rate of more than 99.1%. When γ is set to 1, the hybrid intrusion detection model's detection rate is somewhat greater than DT's, but its effectiveness is minimal. The class 1 SVM model's detection rate rises as c rises, as in the event of an unknown attack. However, in this instance, the class 1 SVM model-based anomaly detection approach is outperformed by the abuse detection method employing DT.



(a) Known assaults with parameter γ=0.01 on ROC Curve



(b) Known assaults with parameter γ=0.1 on ROC Curve



(c) known assaults with parameter γ=1 on ROC Curve

Fig. 7. Investigation and comparison of detection efficiency for known attacks using ROC curves.

This paper introduces an innovative approach to intrusion detection in computer networks by combining abuse detection and anomaly detection techniques. This hybrid approach looks promising because it leverages the strengths of both methods and potentially leads to more effective network security. To select the algorithm, it is also attractive to use computational intelligence algorithms such as frog jumping algorithm, decision trees, support vector machines (SVM) and radial-based neural networks. These algorithms offer a variety of ways to handle intrusion detection, and it makes sense to choose them based on the specific needs of the system. It is also instructive about the execution speed and comparison between SVM and radial neural networks. It is noteworthy to observe that the use of RBF can significantly reduce model training time, as real-time intrusion detection often requires efficient algorithms. Overall, the paper provides a comprehensive overview of the proposed intrusion detection system, its methodology and performance metrics. It provides valuable insights into the potential benefits of combining exploit and anomaly detection techniques and provides a structured approach for future research in network security.

This paper presents a detailed evaluation of the proposed intrusion detection system, including results obtained from experiments using the NSL-KDD dataset. The evaluation includes various performance metrics and comparisons with other intrusion detection systems. Performance measures, detection rate, comparative analysis, training and test results, execution speed and ROC curves are among the evaluations discussed in this research. This paper concludes that the proposed hybrid intrusion detection system provides competitive detection rates and feature selection capabilities compared to other systems. This shows that the system performance can be improved by using RBF models instead of SVM in certain scenarios.

In general, the results and evaluations presented in the paper show the effectiveness of the proposed intrusion detection system in detecting known and unknown attacks. Using various performance metrics, comparative analysis and visual displays (such as ROC curves) provide a comprehensive assessment of system capabilities and tradeoffs. These results help to understand how computational intelligence algorithms can enhance network security.

V. Conclusion

This paper provided a hybrid approach to network intrusion detection. This section will contrast the proposed mixing system's performance with that of comparable systems that employ feature selection algorithms and classification tools from various models, such as decision trees. The proposed mixing system utilizes multiple methods and models in its various components. SVM, Natural Bayes, and artificial neural networks have all been utilized for intrusion detection. The system test with KDD'99 or NSL-KDD data produced the findings that are shown in Table VII. The NSL-KDD dataset, which has the same number of characteristics as the original KDD'99 dataset, should be noted. Classifiers do not favor data with more repetitions since the extension records in the training set are destroyed during the compression process. The majority of detection rates, though, are higher on KDD'99 than NSL-KDD.

The hybrid approach suggested in this work has the best detection rate compared to other models tested on NSL-KDD data in prosperous years, as shown in Table VII. Meanwhile, the proposed models use fewer features than other models, which is a smaller number of features overall. As a result, it can be said that the suggested system offers a good mix of feature selection and classification techniques and that its

performance results are comparable to those of other systems that have been shown effective using KDD'99 and NSL-KDD data.

Additionally, the findings of the experiment demonstrated that while employing parallel SVMs, whose inputs are the leaves of a tree from smaller and more coherent data, resulting in a reduction in model training time, the model execution time relative to using RBF is increased. Instead of SVM, it is significantly longer (roughly 28 times the execution time of the model using RBF), which results in the model's inefficiency, particularly in high-speed networks; In the scenario where the execution errors and false alarm rates of the system are comparable in both models. It can be stated that using RBF instead of SVM has considerably increased the efficiency of the combined model because model training is an offline process, whereas testing and implementation are online processes.

This article examines the performance of the system in detecting known and unknown attacks. However, the system's effectiveness in detecting brand new and zero-day attacks has not been addressed. Future research can focus on developing methods to enhance the detection of previously unseen threats. We also pointed out in this research that using SVM instead of RBF can lead to a significant increase in execution time. Scalability is a critical concern for intrusion detection systems, especially in high-speed networks. Future research can explore ways to optimize the computational efficiency of the system without compromising the detection accuracy. In summary, future research in intrusion detection should focus on addressing these limitations and advancing the field by developing more robust systems. A more consistent and efficient focus can effectively detect a wide range of network intrusions while taking into account ethical and privacy considerations.

For those people who are eager for more research and new work in this field, it is suggested that in order to reduce false alarms in the anomaly detection stage, they should look for a solution to reduce the rate of false negative alarms in the abuse detection stage, because the outputs of the abuse detection stage which were categorized under the title of normal, are used as the input of the anomaly detection stage, and the anomaly detection model needs normal data as its input for better training and detection of violations from the normal pattern, and by reducing the negative rate of error in the detection stage abuse can achieve this.

REFERENCES

[1] V. Kanimozhi and T. P. Jacob, "Artificial intelligence-based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing," in *2019 international conference on communication and signal processing (ICCSP)*, IEEE, 2019, pp. 33–36.

[2] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowl Based Syst*, vol. 64, pp. 22–31, 2014.

[3] S. Khan, K. Kifayat, A. Kashif Bashir, A. Gurtov, and M. Hassan, "Intelligent intrusion detection system in smart grid using computational intelligence and machine learning," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, p. e4062, 2021.

[4] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Syst Appl*, vol. 42, no. 5, pp. 2670–2679, 2015.

[5] A. Shukla, S. Ahamad, G. N. Rao, A. J. Al-Asadi, A. Gupta, and M. Kumbhkar, "Artificial intelligence assisted IoT data intrusion detection," in *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, IEEE, 2021, pp. 330–335.

[6] J. Pirgazi and A. R. Khanteymoori, "SFLA based gene selection approach for improving cancer classification accuracy," *AUT Journal of Modeling and Simulation*, vol. 47, no. 1, pp. 1–8, 2015.

[7] Haoyan Zhang, Xudong Zhao, Huangqing Wang, Ben Niu, Ning Xu, Adaptive Tracking Control for Output-Constrained Switched MIMO Pure-Feedback Nonlinear Systems with Input Saturation, Journal of systems science & complexity, 36: 960–984, 2023.

[8] Ning Xu, Zhongyu Chen, Ben Niu, and Xudong Zhao. Event-Triggered Distributed Consensus Tracking for Nonlinear Multi-Agent Systems: A Minimal Approximation Approach, IEEE Journal on Emerging and Selected Topics in Circuits and Systems, DOI: 10.1109/JETCAS.2023.3277544, 2023.

[9] Arefanjazi, H., Ataei, M., Ekramian, M., & Montazeri, A. (2023). A robust distributed observer design for Lipschitz nonlinear systems with time-varying switching topology. *Journal of the Franklin Institute*, *360*(14), 10728-10744.

[10] G. Folino and P. Sabatino, "Ensemble based collaborative and distributed intrusion detection systems: A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 1–16, 2016.

[11] Fabin Cheng, Ben Niu, Ning Xu, Xudong Zhao, and Adil M. Ahmad. Fault Detection and Performance Recovery Design With Deferred Actuator Replacement Via A Low-Computation Method, IEEE Transactions on Automation Science and Engineering, DOI: 10.1109/TASE.2023.3300723, 2023.

[12] A. A. Aburomman and M. B. I. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Appl Soft Comput*, vol. 38, pp. 360–372, 2016.

[13] A. Deshpande, "A Review on Intrusion Detection System using Artificial Intelligence Approach".

[14] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017.

[15] S. Shamshirband, M. Fathi, A. T. Chronopoulos, A. Montieri, F. Palumbo, and A. Pescapè, "Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues," *Journal of Information Security and Applications*, vol. 55, p. 102582, 2020.

[16] E. Khezri, E. Zeinali, and H. Sargolzaey, "A novel highway routing protocol in vehicular ad hoc networks using VMaSC-LTE and DBA-MAC protocols," *Wirel Commun Mob Comput*, vol. 2022, 2022.

[17] S. R. Islam, W. Eberle, S. K. Ghafoor, A. Siraj, and M. Rogers, "Domain knowledge aided explainable artificial intelligence for intrusion detection and response," *arXiv preprint arXiv:1911.09853*, 2019.

[18] T. Alladi, V. Kohli, V. Chamola, F. R. Yu, and M. Guizani, "Artificial intelligence (AI)-empowered intrusion detection architecture for the internet of vehicles," *IEEE Wirel Commun*, vol. 28, no. 3, pp. 144–149, 2021.

[19] V. Kanimozhi and T. P. Jacob, "Artificial Intelligence outflanks all other machine learning classifiers in Network Intrusion Detection System on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing," *ICT Express*, vol. 7, no. 3, pp. 366–370, 2021.

[20] S. Alzahrani and L. Hong, "Detection of distributed denial of service (DDoS) attacks using artificial intelligence on cloud," in 2018 IEEE World Congress on Services (SERVICES), IEEE, 2018, pp. 35–36.

[21] Wang, Z., Jin, Z., Yang, Z., Zhao, W., & Trik, M. (2023). Increasing efficiency for routing in Internet of Things using Binary Gray Wolf Optimization and fuzzy logic. Journal of King Saud University-Computer and Information Sciences, 101732.

[22] Haoyu Zhang, Quan Zou, Ying Ju, Chenggang Song, Dong Chen. Distance-based Support Vector Machine to Predict DNA N6-methyladine Modification. Current Bioinformatics. 2022, 17(5): 473-482.

[23] FanghuaTang, Huanqing Wang, Liang Zhang, Ning Xu, Adil M.Ahmad. Adaptive optimized consensus control for a class of nonlinear multi-agent systems with asymmetric input saturation constraints and hybrid faults. Communications in Nonlinear Science and Numerical Simulation, 126: 107446, 2023.

[24] E. Khezri and E. Zeinali, "A review on highway routing protocols in vehicular ad hoc networks," *SN Comput Sci*, vol. 2, pp. 1–22, 2021.

[25] Yan, S., Gu, Z., Park, J.H., and Xie, X., 2023. A delay-kernel-dependent approach to saturated control of linear systems with mixed delays. Automatica, 152, p. 110984.

[26] A. Branitskiy and I. Kotenko, "Hybridization of computational intelligence methods for attack detection in computer networks," *J Comput Sci*, vol. 23, pp. 145–156, 2017.

[27] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Computer Networks*, vol. 188, p. 107840, 2021.

[28] T. C. Truong, I. Zelinka, J. Plucar, M. Čandík, and V. Šulc, "Artificial intelligence and cybersecurity: Past, presence, and future," in *Artificial intelligence and evolutionary computations in engineering systems*, Springer, 2020, pp. 351–363.

[29] S. Zhao, S. Li, L. Qi, and L. Da Xu, "Computational intelligence enabled cybersecurity for the internet of things," *IEEE Trans Emerg Top Comput Intell*, vol. 4, no. 5, pp. 666–674, 2020.

[30] M. Samiei, A. Hassani, S. Sarspy, I. E. Komari, M. Trik, and F. Hassanpour, "Classification of skin cancer stages using a AHP fuzzy technique within the context of big data healthcare," *J Cancer Res Clin Oncol*, pp. 1–15, 2023.

[31] J. Sun, Y. Zhang, and M. Trik, "PBPHS: a profile-based predictive handover strategy for 5G networks," *Cybern Syst*, pp. 1–22, 2022.

[32] R. Patgiri, U. Varshney, T. Akutota, and R. Kunde, "An investigation on intrusion detection system using machine learning," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2018, pp. 1684–1691.

[33] M. Trik, H. Akhavan, A. M. Bidgoli, A. M. N. G. Molk, H. Vashani, and S. P. Mozaffari, "A new adaptive selection strategy for reducing latency in networks on chip," *Integration*, vol. 89, pp. 9–24, 2023.

[34] M. Trik, A. M. N. G. Molk, F. Ghasemi, and P. Pouryeganeh, "A hybrid selection strategy based on traffic analysis for improving performance in networks on chip," *J Sens*, vol. 2022, 2022.

[35] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans Emerg Top Comput Intell*, vol. 2, no. 1, pp. 41–50, 2018.

[36] Somanath, S., Wakkary, R., Ettehadi, O., Lin, H., Behzad, A., Eshpeter, J., & Oogjes, D. (2022). Exploring the composite intentionality of 3D printers and makers in digital fabrication. *International Journal of Design*, *16*(3), 77-95.

[37] M. Trik, S. P. Mozaffari, and A. M. Bidgoli, "Providing an adaptive routing along with a hybrid selection strategy to increase efficiency in NoC-based neuromorphic systems," *Comput Intell Neurosci*, vol. 2021, 2021.

[38] D. Mokhlesi Ghanevati, E. Khorami, B. Boukani, and M. Trik, "Improve replica placement in content distribution networks with hybrid technique," *Journal of Advances in Computer Research*, vol. 11, no. 1, pp. 87–99, 2020.

[39] S. Hanif, T. Ilyas, and M. Zeeshan, "Intrusion detection in IoT using artificial neural networks on UNSW-15 dataset," in *2019 IEEE 16th international conference on smart cities: improving quality of life using ICT & IoT and AI (HONET-ICT)*, IEEE, 2019, pp. 152–156.

[40] Khalafi, M., & Boob, D. (2023, July). Accelerated Primal-Dual Methods for Convex-Strongly-Concave Saddle Point Problems. In *International Conference on Machine Learning* (pp. 16250-16270). PMLR.

[41] E. Khezri, E. Zeinali, and H. Sargolzaey, "SGHRP: Secure Greedy Highway Routing Protocol with authentication and increased privacy in vehicular ad hoc networks," *PLoS One*, vol. 18, no. 4, p. e0282031, 2023.

[42] Golrou, A., Rafiei, N., & Sabouri, M. Wheelchair Controlling by eye movements using EOG based Human Machine Interface and Artificial Neural Network. *International Journal of Computer Applications*, *975*, 8887.

[43] I. H. Abdulqadder, S. Zhou, D. Zou, I. T. Aziz, and S. M. A. Akber, "Multi-layered intrusion detection and prevention in the SDN/NFV enabled cloud of 5G networks using AI-based defense mechanisms," *Computer Networks*, vol. 179, p. 107364, 2020.