Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

A Hybrid Three-Layered Approach for Intrusion Detection Using Machine Learning Methods

A. Beigi *

School of Computer Engineering, Shadid Rajaee Teacher Training University, Tehran, Iran.

Article Info

Article History:
Received 08 December 2024
Reviewed 05 February 2025
Revised 18 February 2025
Accepted 09 March 2025

Keywords:
Intrusion detection systems
Network security
Machine learning
NSL-KDD data set

*Corresponding Author's Email Address: *akrambeigi@sru.ac.ir*

Abstract

Background and Objectives: Intrusion Detection Systems (IDS) are crucial for safeguarding computer networks. However, they face challenges such as detecting subtle intrusions and novel attack patterns. While signature-based and anomaly-based IDS have been widely used, hybrid approaches offer a promising solution by combining their strengths. This study aims to develop a robust hybrid IDS that effectively addresses these challenges.

Methods: We propose a three-layered hybrid IDS that leverages machine learning techniques. The first layer utilizes a signature-based approach to identify known intrusions. The second layer employs an anomaly-based approach with unsupervised learning to detect unknown intrusions. The third layer utilizes supervised learning to classify intrusions based on training data. We evaluated the proposed system on the NSL-KDD dataset.

Results: Experimental results demonstrate the effectiveness of our proposed hybrid IDS in accurately detecting intrusions. Comparisons with recent studies using the same dataset show that our system outperforms existing approaches in terms of detection accuracy and robustness.

Conclusion: Our research presents a novel hybrid IDS that effectively addresses the limitations of traditional IDS methods. By combining signature-based, anomaly-based, and supervised learning techniques, our system can accurately detect both known and unknown intrusions. The promising results obtained from our experiments highlight the potential of this approach in enhancing network security.

Introduction

The widespread adoption of the internet has led to a significant increase in data exchange between diverse devices. Ensuring secure communication among these devices is paramount, making network security a critical research area in today's interconnected world. Intrusion Detection Systems (IDSs) play a vital role in bolstering network security, often employed in conjunction with other protective measures such as firewalls and access control mechanisms [1]. Intrusion Detection Systems are designed to safeguard critical resources by actively monitoring network traffic and system events for any signs of unauthorized access. Intruders employ a diverse range of tactics, including [2]:

- Denial of Service (DoS) attacks: These attacks aim to overwhelm a system with excessive traffic, making it unavailable to legitimate users.
- Remote-to-Local (R2L) attacks: These attacks exploit vulnerabilities to gain unauthorized access to a system from a remote location.
- User-to-Root (U2R) attacks: These attacks enable attackers to gain root-level privileges on a system by exploiting vulnerabilities in user accounts.
- Probing attacks: These attacks involve scanning

systems for vulnerabilities that can be later exploited for malicious purposes.

Furthermore, intruders constantly develop new and sophisticated techniques to breach system defenses, making it crucial for IDSs to adapt and evolve.

The performance of an IDS is analyzed by creating a specialized dataset comprising network traffic features to capture and learn attack patterns. Intrusion detection is framed as a classification problem, where various Machine Learning and Data Mining techniques are applied to categorize network data into normal and malicious traffic. The dataset includes both normal and anomalous network traffic, providing the classifier with sufficient examples to identify and differentiate between these patterns effectively [3].

However, some intrusion samples are almost identical to normal samples, leading to false positives where IDS mistakenly classify normal traffic as attacks [4]. To address this issue, researchers often incorporate signature-based methods into their system design. The signature-based method relies on a database of known attack signatures, comparing these signatures with incoming samples to identify intrusions [5].

Another significant challenge in designing an IDS is detecting intrusions that have no prior examples in the training data. To tackle this, researchers employ anomaly-based methods. This approach involves detecting abnormal patterns by monitoring data for deviations from expected behavior. Any significant deviation is flagged as an intrusion [6]. Algorithms utilizing anomaly-based methods can be further divided into supervised and unsupervised learning techniques [7]. Supervised learning algorithms require labeled training data, while unsupervised learning algorithms do not, allowing them to identify new types of intrusions without prior examples.

These challenges can be simultaneously addressed using hybrid approaches that combine signature-based and anomaly-based methods. To maximize the effectiveness of hybrid IDS, attention should be paid to two critical points:

Comprehensive Detection: Hybrid systems should be capable of detecting all types of attacks, including those with training samples (known attacks), those without training samples (unknown attacks), and those that closely resemble normal samples.

Sequence of Methods: The order in which signature-based and anomaly-based methods are applied is crucial and can significantly affect the system's accuracy. For instance, if an anomaly-based algorithm is applied first, it might fail to detect intrusions that are very similar to normal traffic, leading to false negatives. Conversely, applying a signature-based method first can help filter out known attacks, allowing the anomaly-based method to focus on detecting unknown and subtle deviations.

While implementing hybrid methods increases temporal complexity, this approach effectively addresses the outlined challenges and enhances overall system performance.

This study proposes a hybrid intrusion detection system with three detection layers to address the mentioned challenges effectively. Machine learning methods are applied throughout the system. The first detection layer employs heuristic rules, a signature-based approach, to tackle the challenge of intrusions that closely resemble normal samples. The second layer uses a genetic algorithm-based clustering method, an anomaly-based approach, to detect unknown attacks. In the third detection layer, a Back Propagation Neural Network (BPNN) classification is utilized to distinguish known attacks from normal samples [8].

The innovation of the proposed IDS lies in the integration of both supervised and unsupervised learning algorithms alongside a signature-based approach, enhancing its capability to detect a wide range of intrusions. Experiments conducted on the NSL-KDD dataset demonstrate the efficiency of the proposed IDS, showing superior performance compared to other methods using the same dataset.

The remainder of this paper is organized as follows: First, we provide a brief review of recent hybrid approaches for IDS similar to our research. Next, we describe the proposed method, outlining the main idea and the proposed algorithms. Then, we introduce the dataset used and present experimental results demonstrating the efficiency of our method. Finally, we conclude the paper.

Related Studies

Many researchers have developed intrusion detection systems using signature-based, anomaly-based, or hybrid approaches, each aiming to improve the efficiency of their system by leveraging the strengths of these methods. This section reviews some recent studies in this area.

In [7], scholars presented a fuzzy semi-supervised learning (FSSL) method. This approach categorizes an unlabeled training dataset into three fuzzy groups: low, medium, and high, using a neural network classifier. The "low" and "high" fuzzy groups are then combined with a labeled training dataset to form a new training dataset, which is subsequently used to train a neural network. This method enhances the classification efficiency in IDS. The use of a semi-supervised learning method makes it suitable for detecting unknown intrusions; however, it is less effective in detecting attacks that are very similar to normal samples.

In [9], researchers proposed a two-layered hybrid approach. The first layer employs a novel intrusion detection method based on changing cluster centers of samples. The second layer uses the K-Nearest Neighbors

(KNN) algorithm to implement two different detection modules for anomaly and signature-based detection. The second layer's modules reduce the rates of false positives and false negatives from the first detection module. Their experiments show that this method effectively identifies both known and unknown attacks with a high detection rate and low false positive rate, although it struggles with detecting intrusions similar to normal samples.

In [10], a hybrid approach called Hierarchical Filtration of Anomalies (HFA) was introduced. This method first uses a decision tree to separate normal and attack samples. Next, the detected normal samples are re-evaluated by a random forest algorithm, while the detected attack samples are rechecked by the KNN algorithm. Samples classified as normal by KNN are sent back to the random forest for final consideration. This method is effective for detecting known attacks but not for unknown intrusions due to its reliance on supervised algorithms.

Authors in [11] introduced a hybrid IDS based on Artificial Bee Colony (ABC) and Artificial Fish Swarm (AFS) algorithms. Their method preprocesses the training data and divides them into clusters using Fuzzy C-Means Clustering. An initial population is generated for each training subset, and irrelevant features are omitted using Correlation-based Feature Selection (CFS). Trustworthy and straightforward rules are then produced to detect normal and abnormal activities for each subset, which are combined to create the final rules. The ABC-AFS hybrid algorithm is trained on these rules, and a test dataset is used to evaluate system efficiency. However, this method's effectiveness depends heavily on the training dataset, making it less effective for detecting attacks without training samples.

In [12], a multi-level hybrid IDS using Support Vector Machine (SVM) and extreme learning machine was proposed. The method applies the K-means algorithm to create a modified training dataset and employs an anomaly-based approach. Their results indicate limited success in detecting attacks that closely resemble normal samples. Another study, [13], proposed a deep learning-based IDS using Recurrent Neural Networks (RNN-IDS). RNNs can retain previous information and apply it to the current output, making them effective for supervised classification. This method outperforms traditional learning methods like decision trees, SVM, and neural networks but lacks a solution for detecting unknown intrusions.

In [14], a fuzziness-based semi-supervised learning approach via ensemble learning (FSSL-EL) was applied in a framework for IDS. This framework learns from labeled data and analyzes unlabeled and noisy data using a fuzziness-based method. The results of the supervised and unsupervised parts are then combined via an ensemble system. A recently published study, [15], proposed an IDS (TSE-IDS) based on hybrid feature selection and two-level classifier ensembles. This method uses particle swarm optimization, the ant colony algorithm, and a genetic algorithm in the feature selection phase, and rotation forest and bagging in the classifier ensembles stage. The intrusion detection method is anomaly-based, but no specific solution is provided for similar normal intrusions.

In [16], a Deep Convolutional Neural Network (DCNN) model was used for anomaly detection. The model includes an input layer, three convolution and sub-sampling pairs, three fully connected layers, and an output layer with a single sigmoid unit. This research explores the suitability of deep learning approaches for IDS but does not address detecting intrusions similar to normal samples. Another research combines multiple learners to create an ensemble learner called Decision Tree Bagging Ensemble (DTBE) [17]. DTBE first identifies anomalies and then classifies attacks. This method also lacks a specified solution for detecting similar normal intrusions.

In ICVAE-DNN [18], an intrusion detection model is proposed by integrating an Improved Conditional Variational Autoencoder (ICVAE) with a Deep Neural Network (DNN). This model aims to learn and uncover sparse representations between network data features and their corresponding classes. The trained ICVAE decoder generates new attack samples based on specified intrusion categories, effectively balancing the training dataset and enhancing the diversity of training samples. This process improves the detection rate of unbalanced attacks.

In a recent work [19], a hybrid intrusion detection system is proposed. This system leverages the CFS-DE feature selection algorithm for dimensionality reduction and selects an optimal subset of features for improved classification. This system achieves good accuracy by leveraging a diverse set of feature categories during the selection process. It utilizes the CFS-DE algorithm for feature selection.

In [20], an intrusion detection system called IGAN-IDS has been developed to address the challenge of unbalanced class intrusion detection using samples generated by IGAN (Improved Generative Adversarial Network). The system is composed of three main modules: feature extraction, IGAN, and a deep neural network (DNN). Initially, a feed-forward neural network is employed to transform raw network features into feature vectors, which serve as the input for the IGAN module. IGAN then generates synthetic samples to balance the dataset. Finally, a DNN, incorporating convolutional layers and fully connected layers, is utilized for the final intrusion detection, leveraging both the original and the IGAN-generated samples to improve detection accuracy and robustness against class imbalance.

The I-SiamIDS is a two-layer IDS designed for detecting intrusions in unbalanced datasets. This model addresses class imbalance by identifying majority and minority classes at the algorithm level [21]. The first layer employs an ensemble of binary extreme Gradient Boosting, Siamese Neural Network and Deep Neural Network to hierarchically filter input samples and detect potential attacks. Detected attacks are then passed to the second layer, where a multi-class extreme Gradient Boosting classifier further classifies these attacks into specific categories.

CNN-BiLSTM [22] introduces an IDS designed to address class imbalance by combining One-Side Selection with the Synthetic Minority Over-sampling Technique. This hybrid sampling approach enhances model performance by reducing noise in the majority class and increasing the representation of minority class samples. The proposed model leverages Convolutional Neural Networks for spatial feature extraction and Bidirectional Long Short-Term Memory networks for temporal feature extraction. Experimental results on the KDDTest+ dataset demonstrate an accuracy of 83.58%.

Research [23] introduces an intrusion detection method that addresses class imbalance by employing the Adaptive Synthetic Sampling algorithm. To enhance feature extraction and mitigate the impact of redundant information, the model incorporates an improved Convolutional Neural Network based on the Split Convolution Module. This combined approach, termed AS-CNN, is evaluated using the standard NSL-KDD dataset to assess its effectiveness in detecting network intrusions.

BAT-MC [24] presents a two-stage deep learning anomaly detection model that integrates Bidirectional Long Short-Term Memory (BLSTM) networks with an attention mechanism. The attention mechanism effectively prioritizes critical network flow vectors generated by the BLSTM, enabling the model to focus on essential features for precise classification. BAT-MC incorporates convolutional layers alongside BLSTM to capture both spatial and temporal characteristics in network traffic. This end-to-end approach eliminates the need for manual feature engineering, allowing the model to automatically learn hierarchical features. Experimental results on the KDDTest+ dataset demonstrate an accuracy of 84.25%.

Another study [25] proposed a multi-layered approach utilizing k-nearest neighbors (KNN), hyper-learning machines, and hierarchical hyper-learning machines. This work explores the application of Software-Defined Networking (SDN) to mitigate the false positive rate in DoS attack detection systems. The proposed approach combines flow-based and packet-based analysis techniques. Experimental evaluation on the NSL-KDD dataset achieved accuracy of 84.29% on the KDDTest+ subset.

In [26], an Intrusion Detection System (IDS) named DSN was proposed, utilizing a Deep Stacking Network that integrates multiple base classifiers, including decision trees, k-nearest neighbors, deep neural networks, and random forests. The study focuses on the NSL-KDD dataset, with experimental results demonstrating an accuracy of 86.8%.

Recent research [27] explores network intrusion detection by evaluating the performance of various classification techniques, including Decision Tree, Random Forest, Logistic Regression, and K-Nearest Neighbor, on the NSL-KDD dataset. Adopting the CRISP-DM methodology, the study aims to identify and analyze anomalous patterns in network traffic. Among the approaches, the Decision Tree demonstrated the highest performance, achieving complete accuracy on the KDDTrain+ dataset and 80% accuracy on the KDDTest+ dataset.

[28], a recent study, investigates the effectiveness of various shallow machine learning algorithms, including Random Forest, Decision Tree, Gaussian Naïve Bayes, Support Vector Machine, K-Nearest Neighbor, Gradient Boosting, AdaBoost, and Linear Discriminant Analysis, for intrusion detection. This research leverages the NSL-KDD dataset and applies feature selection techniques, such as SelectKBest and Correlation Feature Selection, to enhance model performance. Among the evaluated algorithms, Gradient Boosting achieved the highest accuracy, with 86% in binary classification tasks.

As mentioned earlier, it is crucial that the algorithms are effective in real-world applications and exhibit high performance [17]. A key limitation of many IDSs is their difficulty in detecting unknown attacks and those that closely resemble normal traffic. These limitations significantly reduce both the efficiency and accuracy of such systems. Our proposed method addresses these challenges to improve detection accuracy, and is therefore evaluated using the widely recognized and realistic NSL-KDD dataset. Our system uniquely integrates signature-based detection, genetic algorithm-based clustering, and backpropagation neural networks within a novel three-layered architecture. Experimental results demonstrate that this approach achieves higher detection accuracy compared to other methods evaluated on the same dataset.

Proposed Algorithm

The proposed algorithm comprises two main processes: (1) Clustering and Classification Learning Process Using the Training Dataset, (2) Diagnosis and Separation of Attacks from Normal Samples in the Experimental Dataset. Fig. 1 illustrates the mechanism of the proposed intrusion detection system.

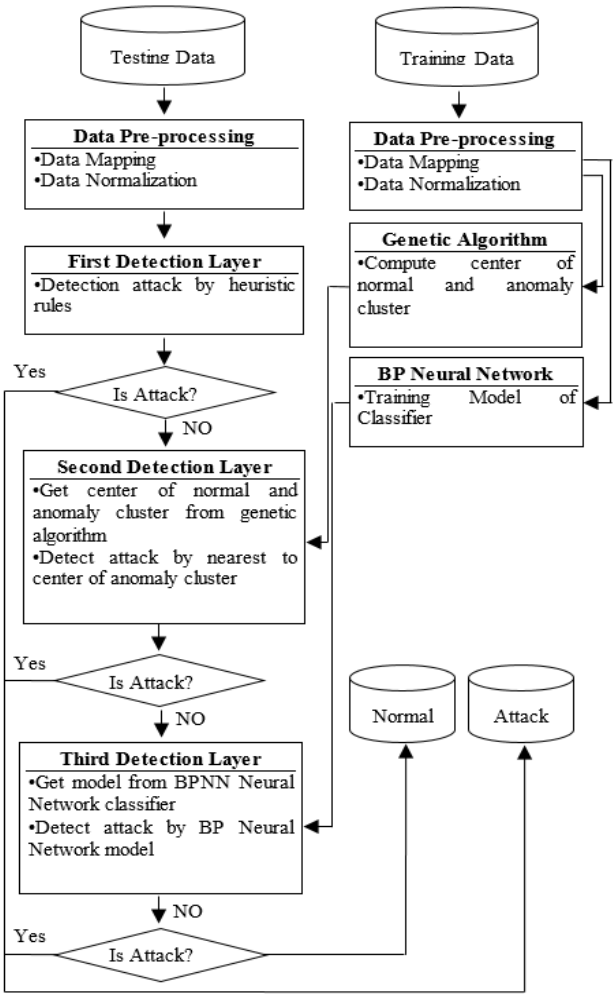

Fig. 1: Proposed intrusion detection system.

The algorithm begins by performing mapping and pre-processing operations on the training and testing datasets. The mapping algorithm involves two processes:

1. Converting discrete feature data to continuous values.
2. Normalizing all values to numbers between zero and one.

For more details on mapping and pre-processing operations, refer to [12]. Next, the training set data is sent to both the genetic-based clustering algorithm and the Back Propagation Neural Network (BPNN) classification algorithm. The genetic-based clustering algorithm divides the data into abnormal and normal subsets and sends these cluster centers to the second diagnostic layer. The BPNN classification algorithm is also trained using this data, and the resulting model is sent to the third diagnostic layer.

In the subsequent phase, each sample in the testing set is processed through all three detection layers for labeling. These three layers utilize detection methods based on heuristic rules, genetic-based clustering, and BPNN classification. Each layer examines the testing samples, and if an anomaly is detected, the sample is labeled as an attack. If none of the layers detect the sample as abnormal, it is labeled as normal. This process continues until all the testing data are labeled. The three detection layers are described as follows:

First Detection Layer: Heuristic Rules

This layer uses a signature-based detection approach with heuristic rules to identify intrusions that closely resemble normal samples.

Second Detection Layer: Genetic-Based Clustering

This layer employs a genetic algorithm-based clustering method to detect unknown attacks by identifying anomalies.

Third Detection Layer: BPNN Classification

This layer uses a Back Propagation Neural Network (BPNN) to classify known attacks and normal samples, providing a final decision on the sample's status.

By integrating these layers, the proposed algorithm addresses the challenges of detecting both known and unknown attacks, as well as intrusions that closely resemble normal behavior. This layered approach enhances the overall accuracy and efficiency of the intrusion detection system.

A. First Detection Layer Based on Heuristic Laws

Many attacks of the Remote to Local (R2L) type exhibit significant similarities with normal samples, making them difficult for classifiers to accurately diagnose. To address this challenge, we implement a detection layer based on heuristic rules. In an intrusion detection system, the expertise of a professional on signature attacks can be translated into heuristic rules in the form of "if-then" statements [4]. By examining the behavior and signature of an attack, we can formulate these rules.

For instance, heuristic rules can effectively detect Guess-password and Warezmaster attacks, which are types of R2L attacks [29]. In Guess-password attacks, the intruder repeatedly attempts different passwords to gain access. If a user fails multiple login attempts in a network event and either succeeds or fails in logging in, such patterns can indicate an attack. Therefore, in datasets where the number of failed logins is recorded in the num-failed-logins attribute and the login status in the logged-in attribute, we can create a heuristic rule based on these attributes.

Warezmaster attacks occur when a file server mistakenly grants write permissions to guest users. During such an attack, an intruder accesses the server using a guest account, creates a hidden folder, and uploads files that can later be downloaded by other users [30]. Heuristic rules can be formulated to detect such activity. The algorithm for this detection layer is illustrated in Fig. 2.

This detection layer uses two heuristic rules:

Heuristic Rule 1: If the number of login failures exceeds one and the user fails to log in, the input sample is detected as an attack.

Heuristic Rule 2: If the protocol is TCP and the service type is FTP or FTP-DATA, the input sample is detected as an attack if either of the following conditions is true:

- The connection time length and the number of bytes sent are greater than zero, and the number of bytes received is zero.
- The guest user has created one or more folders or files in the source.

```
Algorithm. First Detection Layer - heuristic rules
Input:
        Ts_i : Instance Data of Testing Dataset
Output:
        isattack : 0 is not attack and 1 is attack
-----------------------------------------------------------
1:  //rule 1: guess_passwd detection
2:  if (num_failed_logins >0) AND (is_guest_login==0)
3:      isattack=1;
4:      Return;
5:  //rule 2: warezmaster detection
6:  if ((protocol_type == tcp) AND ((service ==ftp)
        OR  (service ==ftp_data )))
7:      if (((duration > 0) AND  (src_bytes >0)
        AND (dst_bytes ==0))
        OR ((hot >0 AND) (is_guest_login ==1)))
8:          isattack=1;
9:          Return;
```

Fig. 2: The algorithm of the first detection layer based on heuristic rules.

These heuristic rules leverage specific attributes and behaviors to distinguish between normal and malicious activities effectively. This layer's ability to detect subtle patterns characteristic of R2L attacks enhances the overall accuracy of the intrusion detection system.

B. Second Detection Layer Based on Genetic Algorithm Clustering

The first detection layer uses heuristic rules to identify several known intrusions, but some intrusions are not detected at this stage. These include both known and unknown intrusions. One of the major challenges of Intrusion Detection Systems (IDSs) is detecting unknown intrusions for which no training examples exist. To address this challenge, a clustering technique can be used to identify unknown attacks. Various clustering algorithms, such as the K-means algorithm and genetic algorithms, have been developed for this purpose. Researchers in [31] utilized a genetic algorithm in their IDS, demonstrating its efficiency over the K-means algorithm based on their results. We have also employed a genetic algorithm in our proposed IDS.

The genetic algorithm is an adaptive and meta-heuristic optimization technique based on the principle of survival of the fittest [32]. In such algorithms, all individuals compete within a generation to acquire resources, and the most successful ones are allowed to produce offspring. Iterations of this approach yield progressively better results with each generation. In this population-based search method, each sample is represented as a chromosome. Chromosomes can be encoded in various forms, including binary, tree structures, permutations, and real values [31]. In our proposed IDS, we use real values for representation. Each generation consists of ten chromosomes, each containing two genes, with each gene comprising 41 alleles. In the proposed algorithm, genes represent the centers of clusters, and alleles denote the characteristics of each center. The initial population is randomly selected from the training dataset, and the quality of each chromosome is evaluated using a proposed fitness function, as shown in Fig. 3. This function helps select the most appropriate chromosomes to produce offspring for the next generation.

```
Algorithm. Fitness of genetic clustering
Input:
        ch: chromosome;
        Tr_d: Data of Training Dataset
Output:
        F_v: Value of Fitness
-----------------------------------------------------------------
1:  set label for all instance of tr_d by nearest to Gen1 OR Gen2;
2:  split tr_d to cluster1 and cluster2 by labels;
3:  compute intra and extra quality of cluster1 AND  cluster2;
4:  set Fitness value(F_v) for qualities;
5:  Return F_v;
```

Fig. 3: The algorithm of the fitness function of genetic clustering,

In this algorithm, one gene on the input chromosome is randomly selected as the center of the normal cluster, and another gene is selected as the center of the abnormal cluster. Then, all samples in the training dataset are labeled based on their Euclidean distance to one of the cluster centers, resulting in two clusters: normal and abnormal. After clustering, new cluster centers are determined using the mean values of the samples within each cluster. Intra-cluster and inter-cluster distances are then calculated as follows:

Intra-Cluster Distance: For each cluster, the Euclidean distance of all samples from the new cluster center is calculated. This value represents the distance within the cluster.

Inter-Cluster Distance: The Euclidean distance between the two new cluster centers is calculated, and a new point is defined at the midpoint of this distance. The Euclidean distance of all samples from this midpoint is then calculated. This value represents the inter-cluster distance.

The fitness function output is higher when the intra-cluster distance is minimized and the inter-cluster distance is maximized. After determining the fitness of all chromosomes, five chromosomes are selected using the Tournament Selection Method. Recombination (at a rate

of 0.8) and mutation (at a rate of 0.01) are performed on them to produce new chromosomes. These new chromosomes are combined with half of the previous generation to form a new generation, and the evaluation process begins again using the fitness function. After 15 generations, the algorithm converges, and in the final generation, the genes of the best chromosomes are considered the centers of the normal and abnormal clusters. These cluster centers are then sent to the second diagnostic layer, where the detection layer labels the experimental input samples based on these clusters. The algorithm for the second recognition layer is shown in Fig. 4.

```
Algorithm. Second Detection Layer - clustering
Input:
        Ts_i : Instance Data of Testing Dataset
        Ts_c1 : center of clusters anomaly
        Ts_c2 : center of clusters normal
Output:
        isattack : 0 is not attack and 1 is attack
-----------------------------------------------------------------------
1:  if (distacnce(Ts_i ,Ts_c1) < distacnce(Ts_i ,Ts_c2) )
2:          isattack=1;
3:  Return;
```

Fig. 4: The algorithm of the second detection layer based on clustering.

In this detection layer, if the input test sample is closer to the center of the abnormal cluster than to the center of the normal cluster, it is labeled as abnormal. This clustering-based approach allows for the identification of unknown intrusions by leveraging the patterns and characteristics inherent in the data.

C. The Third Detection Layer Based on Back Propagation Neural Network

While the first and second detection layers address known and clustered attacks, some attacks may still evade detection due to their similarity to normal samples. To address this challenge and enhance the Intrusion Detection System's (IDS) accuracy, a Back Propagation Neural Network (BPNN) classifier is employed in the third detection layer. The BPNN is a feed-forward neural network used for supervised learning in various applications such as pattern recognition and image processing [8].

The BPNN consists of multiple layers: an input layer, one or more hidden layers, and an output layer. Each layer comprises nodes connected by activation functions, such as hyperbolic tangent or sigmoid functions. During the learning process, the network's settings and connection weights are initially established. For each training sample, a forward calculation is performed from the input layer through the hidden layers to the output layer. A backward calculation follows to correct errors and adjust connection weights. This iterative process enables the network to learn the training samples and recognize unknown patterns effectively. BPNNs have demonstrated high detection rates compared to other neural network techniques [33].

In the proposed IDS, the BPNN classifier is trained using the training dataset. The trained model is then deployed in the third detection layer, where it examines the input experimental samples. If an intrusion is detected, the BPNN classifier labels it accordingly. The algorithm of the third detection layer based on the BPNN is illustrated in Fig. 5.

```
Algorithm. Third Detection Layer - Classifier
Input:
        Ts_i : Instance Data of Testing Dataset
        Net_structure : structure of Back Propagation Neural Network
Output:
        isattack : 0 is not attack and 1 is attack
-----------------------------------------------------------------------
1:  if (distacnce(Ts_i ,Ts_c1) < distacnce(Ts_i ,Ts_c2) )
2:          isattack=1;
3:  Return;
```

Fig. 5: The algorithm of the third detection layer based on BPNN classification.

The third detection layer's algorithm includes the following steps:

Training the BPNN Classifier:

- The classifier is trained using the training dataset.
- Network settings and connection weights are established.
- Forward and backward calculations are performed iteratively to minimize errors and adjust weights.

Deploying the Trained Model:

- The trained BPNN model is sent to the third detection layer.
- The layer uses the trained model to classify input experimental samples.

Intrusion Detection:

- The BPNN classifier examines each input sample.
- If the sample is classified as an intrusion, it is labeled as such.

By incorporating the BPNN classifier in the third detection layer, the IDS can effectively identify intrusions with feature values similar to normal samples, thereby increasing overall detection accuracy. This comprehensive multi-layer approach ensures robust detection of both known and unknown intrusions, leveraging the strengths of heuristic rules, genetic algorithm clustering, and supervised learning.

Experiments and Results

Researchers utilize various datasets to demonstrate the efficiency of their intrusion detection systems. Evaluating an algorithm with real data is crucial as it reflects the algorithm's practical applicability and robustness [17]. In this study, we evaluated the proposed

algorithm using the real-world and widely utilized NSL-KDD dataset. This section discusses the employed dataset, the evaluation methodology of the proposed method, and the experimental results. The proposed algorithms were implemented in MATLAB and executed on a computer equipped with an Intel Core i5 CPU, featuring 4 cores at 2.2 GHz, and 4 GB of RAM.

D. Data Set

The initial event for developing an intrusion detection system took place in 1998, supported by the DARPA organization [3]. During this event, a cyber-attack scenario was simulated at the Air Force Base. This simulation was repeated in 1999 with enhancements by the Computer Security Association [2]. Over seven weeks, raw network TCP/IP data was collected. However, for this data to be useful in learning algorithms, feature extraction was necessary. A team of researchers [3] won the KDD International Knowledge Discovery and Data Mining Tools Competition by presenting a feature extraction method for this raw data set, resulting in the creation of the KDDCUP99 data set. This data set has since become a widely used benchmark in intrusion detection research [7]. In [30], researchers identified several shortcomings in the KDDCUP99 dataset using statistical methods, which significantly impacted system evaluation performance. To address these issues, they proposed an improved dataset known as NSL-KDD, which allows for better comparison of different intrusion detection models.

The NSL-KDD data set maintains the 41 features of the KDDCUP99 data set, and the class labels remain the same. The characteristics of each instance and their details are listed in Table 1 [34].

Most features in the data set have continuous values between 0 and 1, but some are symbolic. Attributes such as land, logged_in, is_host_login, and is_guest_login have values of 0 or 1 and can be treated as continuous properties. The protocol attribute has 3 values, the service attribute has 66 values, and the flag attribute has 11 distinct values that can be converted to continuous values using various methods [12]. Each instance's class determines whether it is normal or an attack. There are several types of attacks in the NSL-KDD database, detailed and classified in Table 2 [7].

The NSL-KDD dataset comprises four subsets: KDDTrain+, KDDTrain+_20%, KDDTest+, and KDDTest-21. The KDDTrain+ and KDDTrain+_20% datasets are used to train learning algorithms, while the KDDTest+ and KDDTest-21 datasets are used to evaluate the performance of intrusion detection algorithms. The training dataset samples are categorized as either normal or known attacks. In contrast, the test dataset includes both known attacks and types of attacks absent in the training dataset, considered unknown intrusions in IDSs. The number of samples for each type in each dataset is shown in Table 3.

Table 1: Description of input sample features [34]

No.	Feature Category	Feature Name	Data Type
1	Basic Features	duration	Continuous
2		protocol_type	Symbolic
3		service	Symbolic
4		flag	Symbolic
5		src_bytes	Continuous
6		dst_bytes	Continuous
7		land	Symbolic
8		wrong_fragment	Continuous
9		urgent	Continuous
10	Content Features	hot	Continuous
11		num_failed_logins	Continuous
12		logged_in	Symbolic
13		num_compromised	Continuous
14		root_shell	Continuous
15		su_attempted	Continuous
16		num_root	Continuous
17		num_file_creations	Continuous
18		num_shells	Continuous
19		num_access_files	Continuous
20		num_outbound_cmds	Continuous
21		is_host_login	Symbolic
22		is_guest_login	Symbolic
23	Traffic Features	count	Continuous
24		srv_count	Continuous
25		serror_rate	Continuous
26		srv_serror_rate	Continuous
27		rerror_rate	Continuous
28		srv_rerror_rate	Continuous
29		same_srv_rate	Continuous
30		diff_srv_rate	Continuous
31		srv_diff_host_rate	Continuous
32	Host-based Features	dst_host_count	Continuous
33		dst_host_srv_count	Continuous
34		dst_host_same_srv_rate	Continuous
35		dst_host_diff_srv_rate	Continuous
36		dst_host_same_src_port_rate	Continuous
37		dst_host_srv_diff_host_rate	Continuous
38		dst_host_serror_rate	Continuous
39		dst_host_srv_serror_rate	Continuous
40		dst_host_rerror_rate	Continuous
41		dst_host_srv_rerror_rate	Continuous

Table 2: Details of attacks type [7]

DoS	U2R	R2L	PROBE
Back	Perl	FTP write	IP sweep
Ping of Death	Buffer Overflow	Guess password	NMAP
Smurf	Load module	IMAP	Port sweep
Land	Rootkit	Multi HOP	Satan
Teardrop		Phf	
		SPY	
		Wareclient	
		Warezmaster	

Table 3: Details of NSL-KDD subsets

Subsets	#Normal data	#Known attack	#Unknown attack	Total
KDDTrain+	67343 53%	58630 47%	0	125973
KDDTrain+_20%	13449 53%	11743 47%	0	25192
KDDTest+	9711 43%	9083 40%	3750 17%	22544
KDDTest-21	2152 18%	5958 50%	3740 32%	11850

According to Table 3, the total number of training samples in KDDTrain+ is approximately five times that of KDDTrain+_20%. Utilizing the KDDTrain+ dataset increases the accuracy of learning algorithms, whereas using the KDDTrain+_20% dataset increases the speed of learning algorithms. Additionally, the total number of samples in the KDDTest-21 collection is about half of those in the KDDTest+ collection, but the number of unknown attacks in both is equal. Therefore, achieving high detection accuracy in an IDS that uses the KDDTest-21 dataset for evaluation is more challenging.

E. Evaluation Criteria

Four states of detection occur for an event in the process of intrusion detection:

True Positive (TP): The input sample is correctly identified as an intrusion at the end of the detection process.

True Negative (TN): The input sample is correctly identified as normal at the end of the detection process.

False Positive (FP): The input sample is actually normal but is incorrectly identified as an intrusion at the end of the detection process.

False Negative (FN): The input sample is actually an intrusion but is incorrectly identified as normal at the end of the detection process.

These four measures (TP, TN, FP, FN) are crucial for evaluating and calculating the accuracy and efficiency of an Intrusion Detection System (IDS). Important criteria for evaluating intrusion detection systems based on these detection modes are defined as follows [12]:

Accuracy (ACC): This metric represents the percentage of correctly labeled samples out of the total samples, as shown in (1):

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Detection Rate (DR): This metric represents the percentage of correctly identified abnormal samples out of the total number of abnormal samples, as shown in (2):

$$DR = \frac{TP}{TP+FN} \quad (2)$$

False Positive Rate (FPR): This metric represents the percentage of incorrectly identified normal samples (false positives) out of the total number of normal samples, as shown in (3):

$$FPR = \frac{FP}{FP+TN} \quad (3)$$

According to these criteria, an IDS performs better when it has higher accuracy (ACC) and detection rate (DR), and a lower false positive rate (FPR) [13], [16].

Experimental Results and Discussion

In this section, we present the results of experiments conducted on the proposed intrusion detection system. We used the KDDTrain+ dataset for training and both KDDTest+ and KDDTest-21 datasets for testing. For evaluating the proposed IDS, all data in the dataset are divided into two classes: normal and attack, and all 41 features of data points have been used in the algorithm's training phase. Table 4 indicates the result of testing performed on the two datasets, KDDTest+ and KDDTest-21, using the proposed IDS.

Table 4 shows the number and ratio of the four parameters: True Positive, True Negative, False Positive, and False Negative. It also shows the False Positive Rate, Detection Rate, and Accuracy of the algorithm.

Table 4: Test results of proposed method on KDDtest+ and KDDtest-21

Subsets	KDDTrain+	KDDTest-21
#TP	10404 46%	7269 61%
#TN	9275 41%	1746 15%
#FP	436 2%	406 3%
#FN	2429 11%	2429 21%
FPR	4.49	18.87
DR	81.07	74.95
ACC	87.29	76.08

As can be seen, the proposed method has achieved significant results. Although many researchers use the NSL-KDD dataset to demonstrate the performance of their intrusion detection systems, few have provided the total accuracy of their method. As mentioned in 2th Section, the following algorithms have good results in the NSL-KDD data set:

- FSSL [7]
- RNN-IDS [13]
- FSSL-EL [14]
- TSE-IDS [15]
- DCNN [16]
- DTBE [17]
- ICVAE-DNN [18]
- CFS-DE_based [19]
- IGAN-IDS [20]
- I-SiamIDS [21]
- SVM [35]

- em J48 [36]
- Two-tier classifier [37]
- Ensemble J48 PART [38]
- CNN-BiLSTM [22]
- AS-CNN [23]
- BAT-MC [24]
- HSDN [25]
- DSN [26]
- CRISP-DM [27]
- Pardeshi [28]

In this section, we compare the total accuracy of the proposed algorithm with that of methods mentioned earlier. We focus on recent works with remarkable results, avoiding comparisons with older studies. Fig. 6 and Fig. 7 display the total accuracy obtained by using the KDDTest+ and KDDTest-21 datasets, respectively.

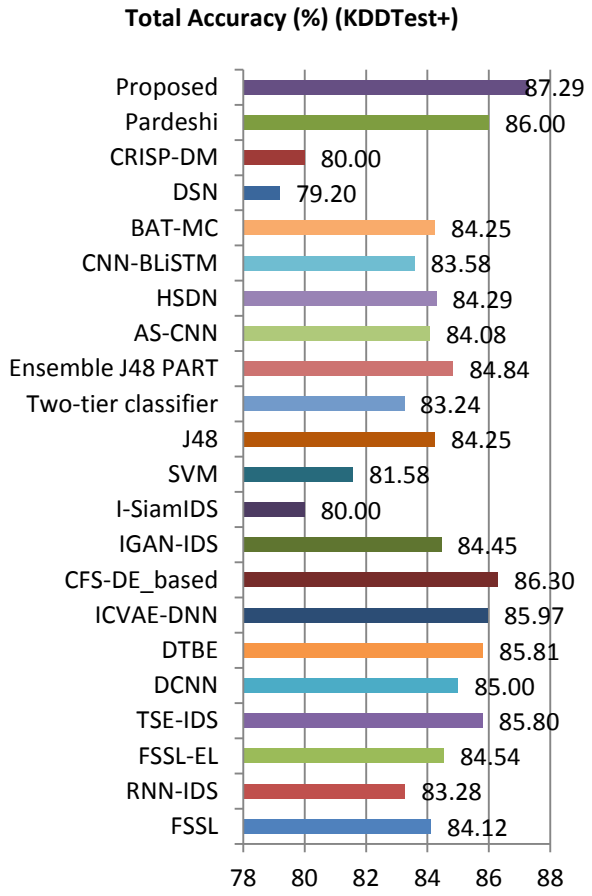

Fig. 6: Comparison of the accuracy of solutions using KDDTest+ data set for binary classification.

It's important to note that the reported results in the CFS-DE_based method, a commonly used approach, are based on testing with all features in dataset. Also, it is worth noting that CFS-DE_based [19], IGAN-IDS [20], I-SiamIDS [21], SVM [35], J48 [36], Two-tier classifier [37], Ensemble J48 PART [38], CNN-BiLSTM [32], HSDN [25], DSN [26], CRISP-DM [27] and Pardeshi [28] have not reported results on KDDTest-21.

The results show that the total accuracy of the proposed method in both experimental datasets is superior to all other mentioned algorithms. As described in 3th Section, this method employs a three-layer hybrid system, with each layer responsible for detecting different types of attacks. Additionally, the sequence of execution of these three layers significantly contributes to the overall success rate and detection of intrusions. As illustrated in Fig. 6 and Fig. 7, the proposed method achieves a higher degree of accuracy. Despite the smaller number of samples and the similar types of attacks in the KDDTest-21 set compared to KDDTest+, the proposed method still maintains high intrusion detection accuracy. Notably, more studies report results on the KDDTest+ database than on KDDTest-21.

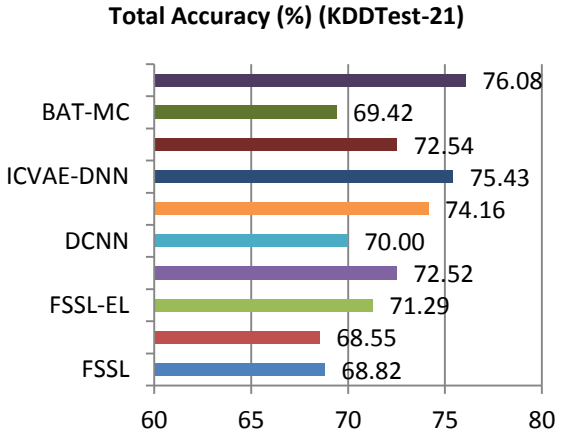

Fig. 7: Comparison of the accuracy of solutions using KDDTest-21 dataset for binary classification.

Evaluating intrusion detection systems requires considering metrics beyond accuracy. As previously mentioned, DR measures the proportion of actual attacks correctly identified, reflecting the system's ability to detect threats, while FPR quantifies the proportion of normal instances incorrectly classified as attacks, directly impacting the number of false alarms. A desirable IDS aims for a high DR and a low FPR. Our proposed method achieved a DR of 81.07% on KDDTrain+ and 74.95% on KDDTest-21, demonstrating strong performance. Our FPR was 4.49% on KDDTest+ and 18.87% on KDDTest-21, representing acceptable results.

It is important to note that many related studies do not report both FPR and DR directly, limiting direct comparison. Among the studies that do, ICVAE-DNN [18] reports DRs of 77.43% on KDDTrain+ and 72.86% on KDDTest-21, while RNN-IDS [13] achieved a DR of 72.95% on KDDTrain+. For FPR, FSSL-EL [14] reported 5.31% on KDDTest+ and 20.35% on KDDTest-21, and RNN-IDS [13] reported 3.6% on KDDTest+, which is lower than our method's FPR on the same dataset. It should be noted that other reviewed studies either used different evaluation metrics or calculated DR and FPR separately for each attack type, making direct comparison in this section inappropriate.

The difference in FPR between KDDTest+ (4.49%) and KDDTest-21 (18.87%) for our method highlights the significant impact of dataset characteristics. The KDDTest-21 dataset, with its higher proportion of unknown attacks, presents a greater challenge for accurate classification. This suggests that while our method effectively handles known attack patterns, further refinement is necessary to improve its ability to generalize to unseen attacks and consequently reduce the FPR in such challenging scenarios.

One crucial aspect to consider when deploying intrusion detection systems in real-world environments is the speed of detection execution. To address this, we have examined the time taken to process an input sample at each of the detection layers individually.

Table 5: The spent time of each detection layer

Detection layer	Detection method	Type of learning	Spent time (S)
Based on heuristic rules	Signature	-	$1.0753*10^{-4}$
Based on clustering	Anomaly	Unsupervised	$3.3016*10^{-4}$
Based on neural networks	Anomaly	Semi-supervised	$614.5862*10^{-4}$

Table 5 indicates that the first and second detection layer does not require much time to diagnose, and their time consuming is close to real-time. The third detection layer is more time-consuming due to the BPNN classifier. It can be said that by using of powerful processors it is possible to implement intrusion detection systems in real-time environments.

Conclusion and Future Work

In this paper, we proposed an intrusion detection system (IDS) employing a hybrid approach to address the challenges of detecting intrusions without training samples and distinguishing between normal samples and those with similar patterns. The system incorporates three detection layers, each employing distinct methodologies to differentiate intrusions from normal samples. The first Detection Layer utilizes signature detection, leveraging heuristic rules to identify known intrusions similar to normal samples. The second Detection Layer is an anomaly-based approach using a clustering method to detect unknown intrusions. The third Detection Layer is another anomaly-based approach using a classification method, specifically a back propagation neural network (BPNN), to detect known intrusions with available training samples.

For evaluation, we utilized the NSL-KDD dataset. By providing solutions for handling similar normal intrusions, intrusions without training samples, and intrusions with training samples, the proposed system demonstrated an increased detection rate and overall accuracy. A comparative analysis of the proposed method's total accuracy against several recently proposed methods evaluated using the NSL-KDD dataset highlights the effectiveness and success of our approach. The hybrid nature of our system, combining signature-based and anomaly-based methods, ensures robust performance across various intrusion scenarios.

Future research will focus on exploring alternative supervised learning algorithms to replace the back propagation neural network in the third detection layer. We anticipate that finding a more efficient algorithm could further enhance the detection rate and accuracy of the proposed IDS, thus continuing to improve its effectiveness in real-world applications.

Author Contributions

A. Beigi designed the experiments, analyzed the data, interpreted the results, and authored the manuscript.

Acknowledgment

Akram Beigi acknowledges the financial support received from Shahid Rajaee Teacher Training University under grant number 5973/15.

Conflict of Interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

IDS	Intrusion Detection System
DoS	Denial of Service
R2L	Remote-to-Local
U2R	User-to-Root
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ACC	Accuracy
DR	Detection Rate
FPR	False Positive Rate
BPNN	Back Propagation Neural Network
KNN	K-Nearest Neighbors
SVM	Support Vector Machine

References

[1] A. Thakkar, R. Lohiya, "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions," Artif. Intell. Rev., 55(1): 453-563, 2022.

[2] S. Venkatesan, "Design an intrusion detection system based on feature selection using ML algorithms," Math. Stat. Eng. Appl., 72(1): 702-710, 2023.

[3] A. Thakkar, R. Lohiya, "A review of the advancement in intrusion detection datasets," Procedia Comput. Sci., 167: 636-645, 2020.

[4] M. Sabhnani, G. Serpen, "KDD feature set complaint heuristic rules for R2L attack detection," in Security and Management, 310-316, 2003.

[5] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, 2(1): 20, 2019.

[6] S. Aljawarneh, M. Aldwairi, M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," J. Comput. Sci., 25: 152-160, 2018.

[7] R. A. R. Ashfaq, X. Z. Wang, J. Z. Huang, H. Abbas, Y. L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," Inf. Sci., 378: 484-497, 2017.

[8] I. Goodfellow, Y. Bengio, A. Courville, "6.5 Back-Propagation and Other Differentiation Algorithms," in Deep Learning, MIT Press, 200-220, 2016. ISBN 9780262035613.

[9] C. Guo, Y. Ping, N. Liu, S. S. Luo, "A two-level hybrid approach for intrusion detection," Neurocomputing, 214: 391-400, 2016.

[10] P. Kar, S. Banerjee, K. C. Mondal, G. Mahapatra, S. Chattopadhyay, "A hybrid intrusion detection system for hierarchical filtration of anomalies," in Inf. Commun. Technol. Intell. Syst., 417-426, Springer, Singapore, 2019.

[11] V. Hajisalem, S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," Comput. Netw., 136: 37-50, 2018.

[12] W. L. Al-Yaseen, Z. A. Othman, M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," Expert Syst. Appl., 67: 296-303, 2017.

[13] C. Yin, Y. Zhu, J. Fei, X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, 5: 21954-21961, 2017.

[14] Y. Gao, Y. Liu, Y. Jin, J. Chen, H. Wu, "A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system," IEEE Access, 6: 50927-50938, 2018.

[15] B. A. Tama, M. Comuzzi, K. H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," IEEE Access, 7: 94497-94507, 2019.

[16] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, K. Han, "Enhanced network anomaly detection based on deep neural networks," IEEE Access, 6: 48231-48246, 2018.

[17] P. Illy, G. Kaddoum, C. M. Moreira, K. Kaur, S. Garg, "Securing fog-to-things environment using intrusion detection system based on ensemble learning," in Proc. 2019 IEEE Wireless Commun. Netw. Conf. (WCNC), 1-7, 2019.

[18] Y. Yang, K. Zheng, C. Wu, Y. Yang, "Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network," Sensors, 19(11): 2528, 2019.

[19] R. Zhao, Y. Mu, L. Zou, X. Wen, "A hybrid intrusion detection system based on feature selection and weighted stacking classifier," IEEE Access, 10: 71414-71426, 2022.

[20] S. Huang, K. Lei, "IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks," Ad Hoc Netw., 105: 102177, 2020.

[21] P. Bedi, N. Gupta, V. Jindal, "I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems," Appl. Intell., 51(2): 1133-1151, 2021.

[22] K. Jiang, W. Wang, A. Wang, H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," IEEE Access, 8: 32464-32476, 2020.

[23] Z. Hu, L. Wang, L. Qi, Y. Li, W. Yang, "A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network," IEEE Access, 8: 195741-195751, 2020.

[24] T. Su, H. Sun, J. Zhu, S. Wang, Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," IEEE Access, 8: 29575-29585, 2020.

[25] M. Latah, L. Toker, "Minimizing false positive rate for DoS attack detection: A hybrid SDN-based approach," ICT Express, 6(2): 125-127, 2020.

[26] Y. Tang, L. Gu, L. Wang, "Deep Stacking Network for Intrusion Detection," Sensors, 22(1): 25, 2022.

[27] Y. Yuliana, D. H. Supriyadi, M. R. Fahlevi, M. R. Arisagas, "Analysis of NSL-KDD for the Implementation of Machine Learning in Network Intrusion Detection System," J. Inform. Inf. Syst. Softw. Eng. Appl. (INISTA), 6(2): 80-89, 2024.

[28] N. G. Pardeshi, D. V. Patil, "Binary and Multiclass Classification Intrusion Detection System using Benchmark NSL-KDD and Machine Learning Models," in Proc. 2024 Int. Conf. Data Sci. Netw. Secur. (ICDSNS), 1-7, 2024.

[29] D. Gümüşbaş, T. Yıldırım, A. Genovese, F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," IEEE Syst. J., 15(2): 1717-1731, 2020.

[30] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE Symp. Comput. Intell. Secur. Def. Appl., 1-6, 2009.

[31] N. B. Aissa, M. Guerroumi, "A genetic clustering technique for anomaly-based intrusion detection systems," in Proc. 2015 IEEE/ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distrib. Comput. (SNPD), 1-6, 2015.

[32] D. Greiner, J. Periaux, D. Quagliarella, J. Magalhaes-Mendes, B. Galvan, "Evolutionary algorithms and metaheuristics: applications in engineering design and optimization," Math. Probl. Eng., 2018.

[33] F. Salo, A. B. Nassif, A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection," Comput. Netw., 148: 164-175, 2019.

[34] P. Mishra, V. Varadharajan, U. Tupakula, E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," IEEE Commun. Surv. Tutorials, 21(1): 686-728, 2018.

[35] Q. M. Alzubi, M. Anbar, Z. N. Alqattan, M. A. Al-Betar, R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimization," Neural Comput. Appl., 2019.

[36] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in Proc. Australas. Comput. Sci. Week Multiconf., 1-6, 2018.

[37] H. H. Pajouh, G. Dastghaibyfard, S. Hashemi, "Two-tier network anomaly detection model: a machine learning approach," J. Intell. Inf. Syst., 48: 61-74, 2017.

[38] N. Paulauskas, J. Auskalnis, "Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset," in 2017 Open Conf. Electr. Electron. Inf. Sci. (eStream), 1-5, 2017.

Biographies

Akram Beigi is currently an Assistant Professor in the Computer Engineering Department at Shahid Rajaee Teacher Training University, Tehran, Iran. She earned her M.Sc. and Ph.D. in Computer Engineering – Artificial Intelligence from Iran University of Science and Technology. Her research interests include machine learning, deep learning, cybersecurity, and multi-agent systems. Her expertise spans various AI-driven applications, particularly in anomaly detection, optimization, biometric authentication, and intelligent decision-making systems.

- Email: akrambeigi@sru.ac.ir
- ORCID: 0000-0003-2268-8734
- Web of Science Researcher ID: ADA-5427-2022
- Scopus Author ID: 36661958200
- Homepage: https://www.sru.ac.ir/en/school-of-computer/akram-beigi/

How to cite this paper:
A. Beigi, "A hybrid three-layered approach for intrusion detection using machine learning methods," J. Electr. Comput. Eng. Innovations, 13(2): 443-454, 2025.

DOI: 10.22061/jecei.2025.11530.811

URL: https://jecei.sru.ac.ir/article_2308.html