ISSN: 1687-1006 p
ISSN: 2682-3640 e

The Egyptian International Journal of Engineering Sciences and Technology

https://eijest.journals.ekb.eg/

Vol. 43 (2023) 88–98

DOI: 10.21608/EIJEST.2022.137441.1156

A Comparative Study of Intrusion Detection Systems Applied To NSL-KDD Dataset

Mina E. Magdy[a*], Ahmed M. Matter[b], Saleh Hussin[a], Doaa Hassan [c], Shaimaa A. Elsaid[a]

[a] *Electronics and Communications Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt*
[b] *Department of computer engineering and artificial intelligence, military technical college*
[c] *Computers and Systems Department, National Telecommunication Institute, Cairo, Egypt*

ARTICLE INFO

Article history:

Received 09 May 2022
Received in revised form
13 August 2022
Accepted 09 September
2022
Available online 09
September2022

Keywords:

Cyber security
Network Security
Intrusion detection systems
(IDS)
NSL-KDD Dataset
Machine Learning
Deep learning.

ABSTRACT

Maintaining network security becomes increasingly difficult due to the increasing computational complexity. To address the security issue, an Intrusion Detection System (IDS) is used to detect traffic abnormalities, protect the network from attacks, and reduce functional and financial losses by alerting the network administrator about those behaviors. Many researchers proposed intrusion detection systems exploiting machine learning and Deep learning techniques in order to enhance the process of intrusion detection. This paper provides a comparative study of the most significant intrusion detection systems that applied machine learning and Deep learning techniques to the NSL-KDD dataset. Experimental results reveal that combining supervised and unsupervised learning algorithms can significantly increase the accuracy of intrusion detection system, also utilizing Feature selection algorithms improves the IDS accuracy with noticeable values.

1. Introduction

Cyber security is a mandatory technology in a world where cyber-attacks are increased exponentially. To ensure that assets are protected from cyber-attacks many security measures must be applied. Cybercrimes cost the world too much loss as it's expected to reach up to six trillion Dollars by the end of year 2021[1], and these losses are horrible for both technology and economy. Firewalls and Antivirus prove that its capabilities are limited against many types of Network attacks hence the Intrusion Detection System (IDS) proves that it has a promising result against intrusion attempts and it has a critical security component that can't be neglected. When Cyber-attack takes place, we need to detect it and find detailed information about that attack. For the sake of protecting Information and Communications Technology (ICT) infrastructure, these two important and sorely needed advantages are utilized. The benefits of using IDS are back to its mode of operation. IDSs are classified according to detection schemes to misuse detection and anomaly detection [2]. Misuse detection depends on the signature of the security attacks to detect them. Moreover, it can't detect new attacks because its signature is not available for the IDSs. However, it provides high level of accuracy in recognizing known attacks by its signature. The anomaly detection IDS can detect new attacks by depending on network traffic behavior. In order to accomplish optimum accuracy, anomaly detection system needs training to be able to differentiate between normal and abnormal behavior. To carry out this learning phase, a security dataset is needed. If the training of the anomaly detection system goes well, it can detect and predict new attacks and zero-day attacks. This gives ICT the opportunity to develop appropriate countermeasures to protect its assets.

1.1 Problem Statement

As mentioned above, the increased attacks problem needs more focus so researchers' worldwide try to develop intrusion detection system that can extenuate it. Here we try to perform a comparative study for the latest research in the last three years related to anomaly IDS which applying machine learning and deep learning techniques on NSL-KDD dataset. Our primary target is to work on NSL-KDD dataset, as it is the most well-known benchmark in cyber security.

* Corresponding author. Tel.: +2-01229673504
E-mail address: memw22@gmail.com

1.2 Our Contribution

In this paper we introduce a comparative study of multiple research papers related to anomaly detection using Network Security Laboratory Knowledge Discovery and Data Mining (NSL-KDD) Dataset [3].

The remainder of this paper is organized as follows. In Section 2, Background and Related work are stated. The Subjects and Methods are provided in Section 3. In Section 4, the intrusion detection systems based on NSL-KDD are presented in detail. In Section 5, a performance comparison is provided. At the end of this article, Section 6 provides a conclusion and outlines our future work.

2. Background and Related work

In this section we are discussing different types of IDSs depending on its deployment, response and detection; then related work is discussed.

2.1 Background

Intrusion Detection System aims to detect and correctly classify intrusions and attacks which take place on a network or a host. When it focuses on a network, it's called Network Intrusion detection system (NIDS), and when it focuses on a host it is called Host-Based Intrusion detection system (HIDS) [4]. The aforementioned category is based on the source of data[5][6]. Also, IDS can be classified by its action into passive IDS that log and notify for intrusion, and active that take action and modify environment [7]. The third category of IDS can be classified into signature, anomaly based and stateful protocol analysis. The knowledge based known as signature based is effective with known attacks, anomaly based is effective with unforeseen attacks, and final one specification based which also known as stateful protocol analysis that knows and trace protocol states [8]. In order to choose the right IDS for each environment, the purpose of using IDS must be clear. Intrusion detection system types can be categorized by target, deployment location, approach, structure and response [9] [10] as shown in Fig. 1.

2.2 Related work

There was a research work presented in [11] that proposed a deep learning detection system based on Deep Neural Network (DNN) for software define network choosing only six basic network features (duration, protocol_type, src_bytes, dst_bytes, count and srv_count) from the NSL-KDD dataset with accuracy of 75.75% for anomaly detection. Combining Non-symmetric Deep Auto-Encoder (NDAE) with Random Forest (RF) was proposed in [12] which lead to accuracy improvement upon the accuracy of existing techniques till 2018 to be 89.22% and time saving up to 98.81% using NSL-KDD 13 for anomaly detection.

While in [13] research converts the vector format raw traffic into a format of image data. Afterwards the authors used Convolutional Neural Network (CNN) intrusion detection model to improve accuracy compared to the existing Machine Learning based approaches to be 79.48%. In [14] research proposed a Hierarchical Combining of Predictions of a Tree of Classifiers (HCPTC-IDS) then compare performance with different techniques applied like NB, FL, RIPPER, DT, ANN, and SVM. The proposed model processes each record 373 microseconds that prove its fast processing of data traffic of the NSL-KDD Dataset, with accuracy 89.75%. Combine Sparse Auto-Encoder (SAE) with SVM in [15] named this approach Self-Taught Learning IDS (STL-IDS), Self-Taught Learning (STL) is used for data representation while SVM is used for the classification. The proposed deep learning intrusion detection approach gives accuracy 84.96%. Introducing Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) in [16] that can improve anomaly detection rate by exhibit the relationship between previous and current events, GRU-RNN used in Software Define Network (SDN) environment using minimum number of features (six basic features) achieving 89% accuracy. Introduce Feature Extraction Unit (FEU) in [17] which are based on filter-based algorithms. They also introduced a Feed-Forward Deep Neural Network (FFDNN) as a classifier. It was also shown that, the number of neurons utilized by FFDDN classifier directly impacts the intrusion system accuracy. Using this technique raises the accuracy to be 87.74% with 30 nodes and 3 hidden layers.

The research work presented in [18] proposed an adaptive ensemble learning model that used NSL-KDD data set, and algorithms such as random forest, decision tree, and Deep Neural Network (DNN) to train that model. The proposed adaptive ensemble learning model can achieve accuracy 85.2%. Proposing A Scalable Hybrid Intrusion Detection Alertnet SHIA framework in [2] that process network-level and host-level records. It also proposed a DNN to detect cyber-attacks. Experimentation conducted using NSL-KDD. Accuracy result for Binary class classification 80.1% with one-layer DNN, Multi-class classification 78.5% with 5 layers DNN. Proposing the Improved Conditional Variational Auto-Encoder DNN (ICVAE-DNN) model in [19] using NSL-KDD to experiment that model then compare its accuracy, detection rate and false positive rate with another six classification algorithms: KNN, Multinomial-NB, RF, SVM, DNN and DBN. It was shown that its accuracy is higher than them. Accuracy result for ICVAE-DNN is 85.97%. Self-Adaptive and autonomous misuse IDS was proposed in [20] which depends on Self-taught learning alongside a methodology named MAPE-K. Self-

taught learning is a deep-learning technique able to identify unseen attacks through unlabeled data reconstruction, it can be used with MAPE-K reference model to identify unseen attacks which lead to an accuracy of 77.99%.

Introduce Auto-Encoder (AE) and statistical analysis model, using the NSL-KDD dataset in [21]showing the result of increasing/decreasing hidden layers impact on accuracy with single Hidden Layer HL of 50 units to be Binary classification 84.24%, and for multi-Classification 87%. Presenting a Deep learning system in [22] that gave a high accuracy compared with previously developed systems using Spark Cluster configuration. The proposed system is called DLS-IDS (Deep Learning Spark-IDS). Long-Short Term Memory (LSTM) with Synthetic Minority Over-Sampling Technique (SMOTE) improved the detection accuracy to reach 83.57%. Propose Difficult Set Sampling Technique (DSSTE) algorithm [23] that improved the classification model to empower imbalanced dataset data learning, and used the NSL-KDD as benchmark dataset. They achieved an accuracy of 82.84%. Confirming that there is a direct apposition between accuracy of detection and quality of data collection in [24] research proposed a 5-layer Auto-Encoder model which offer better and accurate identification of anomaly network traffic. The proposed approach was tested on the NSL-KDD dataset, which obtained higher performance of 90.61% accuracy. Proposing a hybrid machine learning model and a feature selection method in [25] that were applied to NSL-KDD dataset with seventeen features selected. The accuracy result measured for the proposed model was 90.41% that showed its performance to be higher than other learning models with 11% and with a higher accuracy and detection rate. Combining deep neural networks which accelerates data processing by utilizing the ReLU activation function with principal component analysis (PCA) in [26] achieve accuracy 88.64 %.proposing enhanced random forest and synthetic minority oversampling technique (SMOTE) algorithm in [27] can achieve accuracy 78.47 %. Proposing multi-module integrated intrusion detection system (GMM-WGAN) Intrusion detection system contain of three parts, feature selection, imbalance processing , and classification in [28] achieved 86.59% accuracy. Proposing model consists of a deep neural network (DNN) trained using 28 features of the NSL-KDD dataset in addition using feature scaling mechanism in [29] achieving 81.87% accuracy.

3. NSL-KDD Dataset

In 1999 one of the most widely used research datasets for cyber security was created named KDD99 [30]. After years of research on KDD99, researchers discovered some disadvantages that need to be solved such as redundancy and the unreasonable number of records in both train and test datasets which make it difficult to work with entire dataset in experiments. To overcome the previously aforementioned disadvantages, a newer version was proposed in [3], namely NSL-KDD. Since 2009, NSL-KDD has been considered as the new benchmark dataset for cyber security research. NSL-KDD dataset includes KDDTrain+ which consists of 125,973 and KDDTest+ which consists of 22,544 records. Each record is represented by 41 features, belonging to four different feature categories including [31]: Basic features, time-based Traffic features, connection-based Traffic features, and Content features. Twenty-one predicated label class for each record to represent attack and normal record. In cyber security community, each record is considered as a session (is connection between two pairs) between two hosts in the network. The probability distribution of KDDTrain+ is not the same of KDDTest+. For test dataset it contains some attacks which not included in training data. Hence the training dataset contain 24 different types of attacks, on the other hand testing dataset contain additional 14 types of attacks do not present in training set to test the ability of classifier to detect unknown attacks. Altogether NSL-KDD furnishes a new idea that improves KDD99. For example, KDD99 considers probing as an attack, while contrary NSL-KDD does not consider it as an attack barring number of iterations surpass a specific threshold. Table 1 describes the NSL-KDD record Details.

Table 1. NSL-KDD record Details

	All Records	Normal	DOS	Probe	R2L	U2R
KDDTrain+	125,973	67343	45927	11656	995	52
KDDTest+	22,544	9711	7458	2421	2754	200

4. Intrusion Detection Systems Based on NSL-KDD

ML and DL are AI branches that use supervised, unsupervised, and reinforcement learning methods to learn. Some points may be used to describe the differences between ML and DL [32]. First, while ML can build a good model with a little or medium quantity of data, DL works better with a huge amount of data. As data increases, DL improves, but growing data with ML does not enhance performance. Hardware dependencies are the second point to consider. While ML can run on a CPU, DL requires a GPU and more powerful hardware to function correctly. The third point is feature processing which is essential for ML intervention to define the correct input; however, DL can learn from data, so it does not require intervention. Fourth point: Problem-solving while DL solves problem in a single step utilizing its many layers, whereas ML divides problem into sub-problems and solves them one at a time [33]. The fifth point is the execution time. In training, DL takes longer than ML; however, in testing, DL becomes quicker, and there are still some algorithms in ML that are faster than DL. In this section we will give a simple summary about some of ML then DL algorithms, starting with ML as follows:

- *Naïve Bayes*

It is one of the simplest algorithms from implementation point of view [25], taking in consideration Naive Bayes ability to build classifiers, this leads to define a model that able to classify any problem; giving them class labels in order to represent it as feature values [8]. When Naive Bayes Classifier is utilized, valuable information about features correlation is determined. The presence of a particular feature in a class is unrelated to the presence of other features [34]. It provides a principled way for calculating the conditional probability. The form of the calculation for Naïve Bayes is as follows

$$P(A|B) = P(B|A) * P(A) / P(B) \qquad (1)$$

Where P(A|B) is called the posterior probability and the P(A) is called the prior.

- *K-means Clustering*

Data is categorized using the K-means clustering algorithm technique. That is, each data point can only be assigned to one of the groups. It concentrates on the ability to discover patterns within supplied data [7]. Anomaly detection may be accomplished using the K-means clustering technique. It is easy to put into action. When there are a lot of overlapping data, K-means clustering efficiency suffers. It is calculated as follows:

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} (|| x_i - v_j ||)^2 \qquad (2)$$

where, $|| x_i - v_j ||$ is the Euclidian distance between a point, x_i , and a centroid v_j , iterated over all *k* points in the i^{th} cluster, for all *n* clusters.

- *Support Vector Machine (SVM)*

This is a classical Machine Learning algorithm for Networking. It is suitable for Machine Learning since it can deal with both classification and regression difficulties. It can also work effectively with little amounts of data. SVM generates a line that divides data into two classes. The goal of this line is to maximize the margin between two classes. In our scenario, it can distinguish between legitimate (normal) and malicious traffic. Large datasets, on the other hand, require longer time to train [34]. In case of classes are linearly separable, we Suppose we are given a dataset (x_1, y_1), (x_2, y_2), ..., (x_m, y_m) where y_i=−1 for inputs x_i in class 0 and y_i=1 for inputs x_i in class 1. In that condition the negative classification boundary is $\vec{w} * \vec{x_n} + b = -1$ and the positive classification boundary is $\vec{w} * \vec{x_p} + b = 1$.

-*Decision Tree*

Decision Tree is a basic machine learning technique that can tackle classification and regression problems, similar to SVM [35]. A Decision Tree is a flowchart-like structure in which each node indicates a test on a single feature. Each leaf is a node that represents a class label, and feature conjunctions indicate branches that lead to those class labels. As pathways from the root to the leaf, the classification rules are well defined [36]. The information gain function formula states the information gain as a function of the entropy of a node of the decision tree as follows:

$$IG (s) = H (t) + H (s,t) \qquad (3)$$

On each iteration of the Decision Tree algorithm, it iterates through the very unused attribute of the set (*s*) to calculate Entropy (*H*) and Information gain (*IG*) of this attribute.

- *Random Forest (RF)*

It's a learning algorithm that's supervised. It can tackle classification and regression problems in the same way as SVM and Decision Tree can[35]. It constructs many decision trees before merging them to produce a more stable and accurate prediction. In compared to SVM, RF takes less time to train. It is also easier to implement, resulting in improved modeling. When it comes to feature significance, RF can calculate the relative value of each feature in terms of prediction. In addition, RF prevent over-fitting problem that reduce generalization performance of classifiers [34]. Using Random Forest Algorithm to solve regression problems, we use the mean squared error (MSE).

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2 \qquad (4)$$

Where N is number of data points, y_i is the value of the data point at a certain node, and f_i is the value returned by the tree.

- *K-nearest Neighbors Algorithm (KNN)*

KNN is a machine learning algorithm that is known to be simple and easy to implement. It performs data classification based on how the nearest data points are. KNN considers the number of nearest neighbors is the variable *k*. As the number of nearest neighbors increased, the accuracy might increase [37]. KNN does not require data preprocessing, training happen with the algorithm memorizing the data. The k-nearest neighbor classifier fundamentally relies on a distance metric. Here is the equation to calculate distance (*d*) for nearest Neighbor:

$$\text{d (x,z)} = (\sum_{r=1}^{d} \mid x_r - z_r \mid^{p})^{1/p} \qquad (5)$$

where *x* is start point, *z* is the nearest neighbor point, and *p* is probability distribution.

- *Artificial Neural Network (ANN)*

Artificial Neural Network is a subfield of machine learning. The ANN aims to provide machine learning system that is based on the biological model of human brain. ANNs are formed of multiple layers [35]. It is like that input layer, hidden layer, and output layer. When it comes to this input layer, each feature in dataset is represented by neuron. Input passes to the next layer. For the Hidden Layer it is considered as a set of neurons, a weight is assigned to each neuron. Every layer takes input from previous layer. Final result is given from output Layer. It is considerable that Neural Networks need more power. There are three steps to perform in any neural network:

- Take the input variables x_i and the linear combination equation of $Z = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$ to compute the output or the predicted *Y* values (Y_{pred}).
- Then calculate the loss or the error term. The error term is the deviation of the actual values Y_{actual} from the predicted values Y_{pred}.
- Minimize the loss function or the error term.

- *Convolutional Neural Network (CNN)*

Another kind of ANN is the convolutional neural network. CNN is most commonly used for image recognition, but it may also be utilized for other tasks, such as Network Intrusion Detection System (NIDS), because it can extract features from network traffic records (connections). CNN takes a long time to train. Its efficiency in calculation, on the other hand, has been demonstrated. It was also shown that CNN is quite effective in feature extraction [38]. However, when compared to other Neural Network methods, CNN is more complicated to construct. For convolutional neural network, the number of output features in each dimension can be calculated by $n_{out} = \left(\frac{n_{in}+2p-k}{s}\right) + 1$. Where n_{in} *is* number of input features, n_{out} *is* number of output features, k convolution kernel size, p convolution padding size and *s* convolution stride size.

- *Auto-Encoder*

Auto-encoder is considered unsupervised neural network and a type of Artificial Neural Network (ANN). Its function is extracting features and dimension reduction. Auto-encoder have layers which are input layer, output layer, and a hidden layer like ANN. Auto-encoder contains encoder and decoder, where encoder takes input data to map it into a code then decoder assigns the code to input data [39]. Auto-encoder can be represented like:

$$\Phi: X \longrightarrow F$$
$$\Psi: F \longrightarrow X$$
$$\Phi, \Psi = arg_{\Phi,\Psi}\min \mid\mid X - (\Phi \circ \Psi)X \mid\mid^2 \qquad (6)$$

Where Φ is the encoder function, *X* original data, *F* latent space which is presented at the bottleneck, Ψ is the decoder function which maps the latent space *F* at the bottleneck to the output. The output, in this case, is the same as the input function.

5. Performance Comparison

5.1 Evaluation Metrics

- Accuracy is a metric which describes the percentage of correctly predicted instances against the original (true) label. The higher the accuracy, the more accurate the generated predictive model is. In cyber security, researchers aim to increase the accuracy of their prediction model to detect normal or attack instances in computer networks. Accuracy is calculated as the number of all correct predictions divided by the total number of the dataset. The best accuracy is 1.

$$\text{Accuracy} = (TP + TN) / (TP + FP + TN + FN) \qquad (7)$$

Where *TP* is true positive, *TN* is True Negative, *FP* is False positive, and *FN* is False Negative.

- Recall (sensitivity) is the percentage of attack instances which classified correctly. Recall can be calculated as the number of correct positive predictions divided by the total number of positives. It is also called true positive rate (TPR). The best sensitivity is 1.

$$\text{Recall} = TP / (TP + FN) \qquad (8)$$

Precision that is the proportion of predicted attack instances which is truly attack instances. It can be calculated as the number of correct positive

predictions divided by the total number of positive predictions. It is also called positive predictive value (PPV). The best precision is 1.

$$\text{Precision} = TP / (TP + FP) \quad (9)$$

For recall and precision, higher value is better. In order to take the benefits of recall and precision combined into one measure the F1-Score metric is used, It is a harmonic mean of precision and recall, calculated as follows

$$\text{F1-Score} = 2TP / (2TP + FP + FN) \quad (10)$$

- When normal instances are detected as attack the metric that can address this issue is False positive rate, calculated as the number of incorrect positive predictions divided by the total number of negatives. The best false positive rate is 0.0 its equation

$$FPR = FP / (FP + TN) \quad (11)$$

5.2. Performance Comparison

Table 2 provides a performance comparison among the previously reviewed IDSs. The common criteria between all are using NSL-KDD dataset and accuracy as an evaluation metric. Figure 2 shows the accuracy of multiple IDSs on NSL-KDD. It is obvious that [24] has the highest accuracy while [11] has the lowest accuracy. Figure 3 shows a precision comparison among the aforementioned IDSs. It is clear that the IDS proposed in [19] has the highest precision. Recall results are shown in Fig. 4. Figure 5 illustrates the F1-score results. For papers which measured the execution time, the platform specification is as follows.

- For [14] the experiments were done on Weka Data Mining Tool installed on Windows PC with 8 GB RAM and CPU I5 1.7 GHZ.
- For [15], Experiments are performed on a Windows 10 PC with Intel(R) Core (TM)i5-6400 CPU at 2.71GHZ with 8 GB of RAM.
- For [26] experiments were performed using Python3.5, and the software package used is sklearn build on Windows10 OS installed on Intel CoreTMi7-9750H CPU, 16GB RAM.

5.3. Discussion

Although Deep learning (DL) is a subset of machine learning (ML) and it works in the same manner as machine learning, it is a better and more advanced one. DL techniques are the best for detecting the intrusions in imbalanced network since it has stronger fitting and generalization abilities. DL techniques are independent of feature engineering and domain knowledge, which is considered an advantage over ML techniques. However, the execution time of DL is often too long to meet the Realtime requirement of IDSs

Table 2. Performance comparison among IDSs on NSL-KDD dataset

Ref.	Approach	Accuracy	Precision	Recall	F1-score	False positive rate	specificity	Testing Time(sec)
[11]	Deep Neural Network (DNN)	75.75	83	76	75	-	-	-
[12]	Non-symmetric Deep Auto-Encoder (NDAE) (deep learning) and the accuracy and speed of Random Forest (RF)	89.22	92.97	89.22	90.76	10.78	-	-
[13]	Convolutional neural network (CNN)	79.48	23.4	68.66	-	27.90	-	-
[14]	Intrusion Detection System based on Hierarchical Combining of Predictions of a Tree of Classifiers (HCPTC-IDS)	89.75	-	86.71	-	6.23	-	8.41
[15]	self-taught learning (STL) +	84.96	96.23	76.57	85.28	-	-	4.648

	SVM							
[16]	A Gated Recurrent Unit Recurrent Neural Network (GRU-RNN)	89	91	90	90	-	-	-
[17]	A Feature Extraction Unit (FEU) and Feed-Forward Deep Neural Network (FFDNN)	87.74	-	-	-	-	-	-
[18]	ADAPTIVE ENSEMBLE LEARNING	85.2	86.5	85.2	84.9	-	-	-
[2]	Deep neural network (DNN), Hybrid intrusion detection framework called SHIA	80.1	69.2	96.9	80.7	-	-	-
[19]	Improved Conditional Variational AutoEncoder (ICVAE-DNN)	85.97	97.39	77.43	86.27	2.74	-	-
[20]	Self-taught learning alongside with MAPE-K (self-adaptive system)	77.99	-	60.34	-	0.4	-	-
[21]	Autoencoder (AE)	84.24	87	80.37	81.98	0.4	-	-
[22]	DLS-IDS (Deep Learning Spark Intrusion Detection System)	83.57	96.46	78.12	86.32	3.57	96.43	-
[23]	Difficult Set Sampling Technique (DSSTE)+alexnet	82.84	83.94	82.78	81.66	-	-	-
[24]	5-layer Autoencoder (AE)	90.61	86.83	98.43	92.26	-	-	-
[25]	Light GBM combined with K-Means	90.41	84.78	86.9	90.96	-	97.8	-
[26]	DT-PCADNN	88.64	-	84.56	-	-	-	57.86
[27]	Enhanced random forest	78.47	-	78	-	-	-	-
[28]	GMM-WGAN	86.59	88.55	86.59	86.88	-	-	-
[29]	RT-IDS	81.87	96.45	70.71	81.59	-	-	-

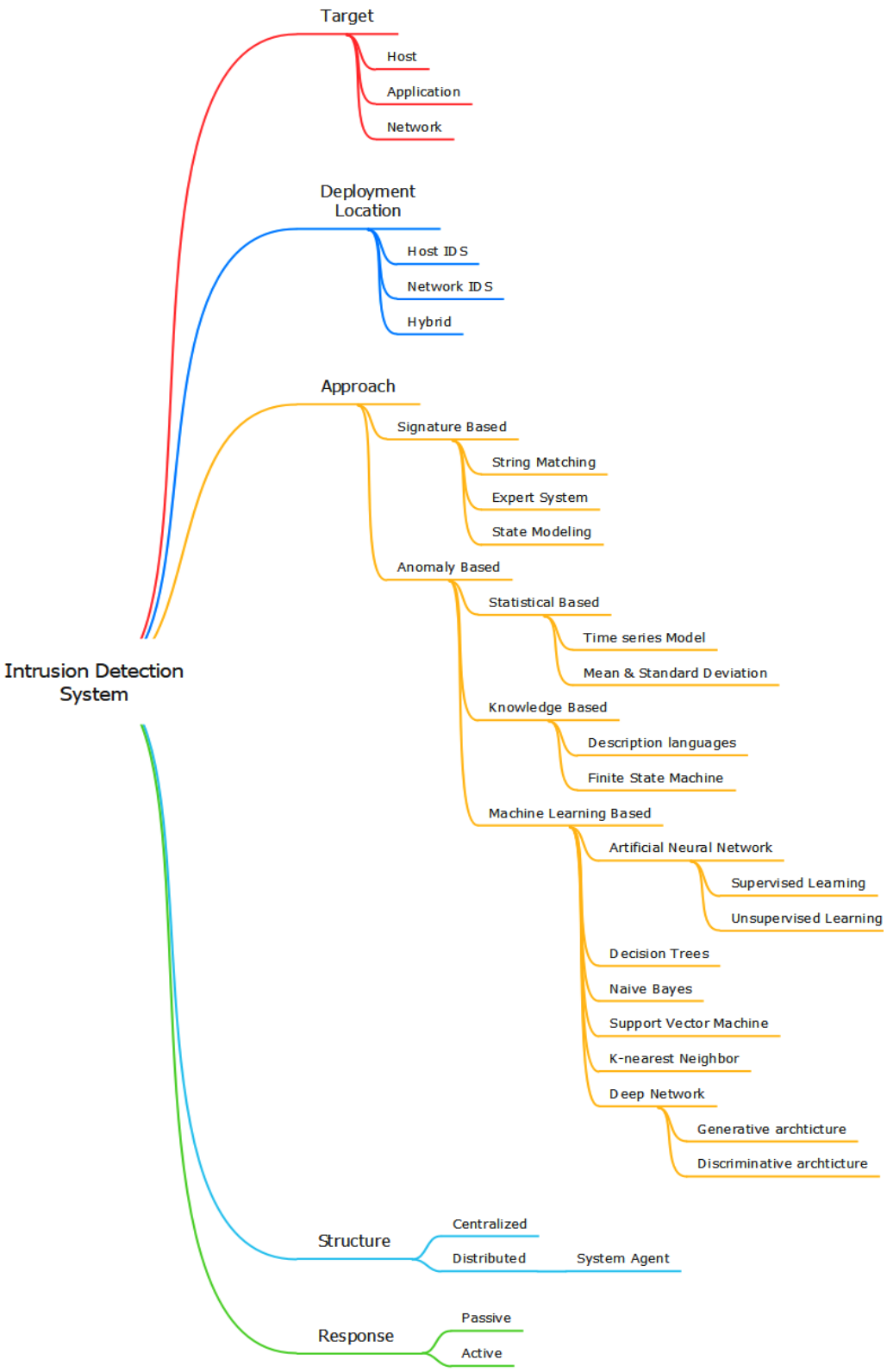

Figure 1. Intrusion detection systems categorized by target, deployment location, approach, structure and response

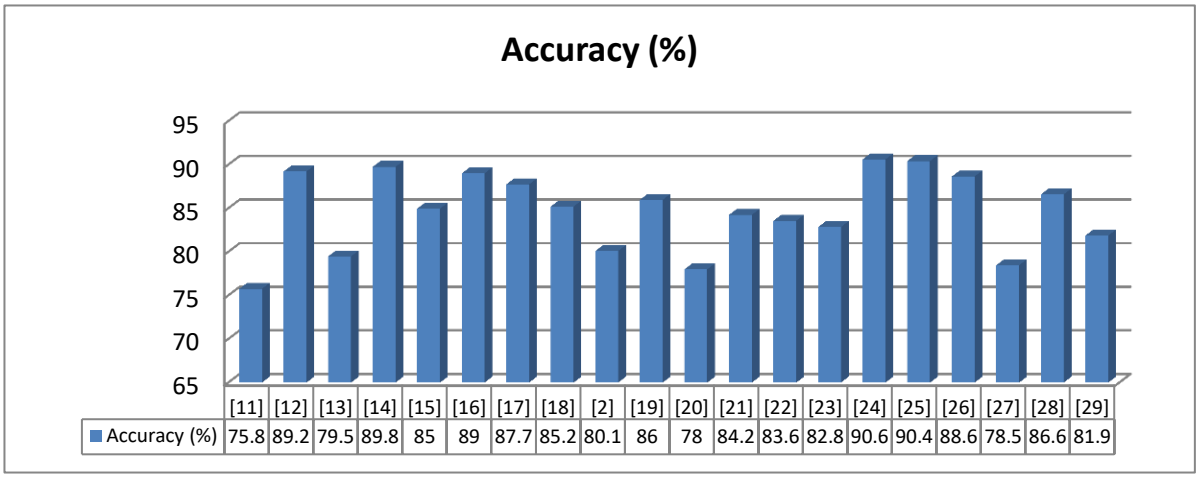

Figure 2. Accuracy Results of multiple IDSs on NSL-KDD

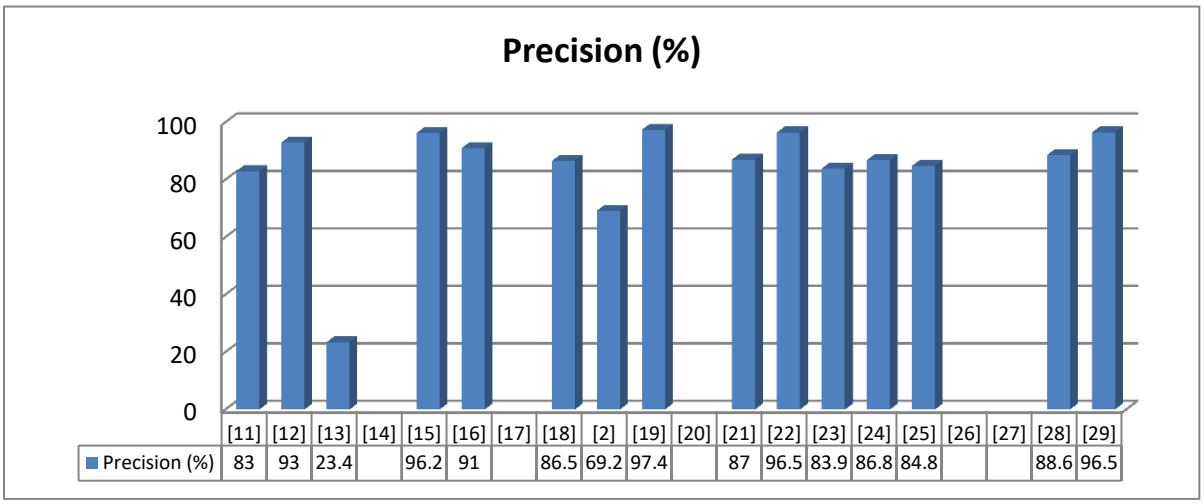

Figure 3. Precision of multiple IDSs on NSL-KDD

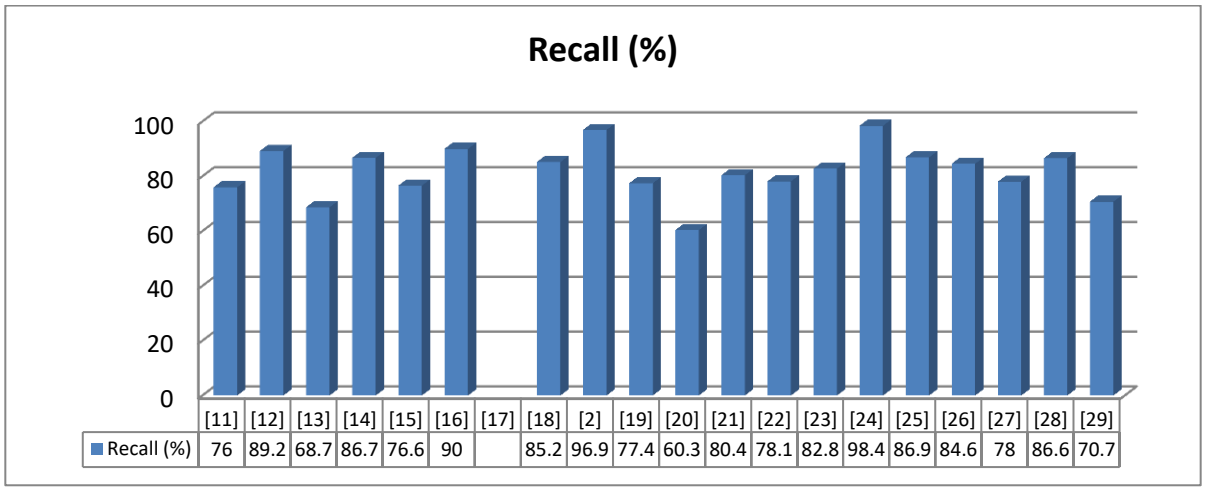

Figure 4. Recall of multiple IDSs on NSL-KDD

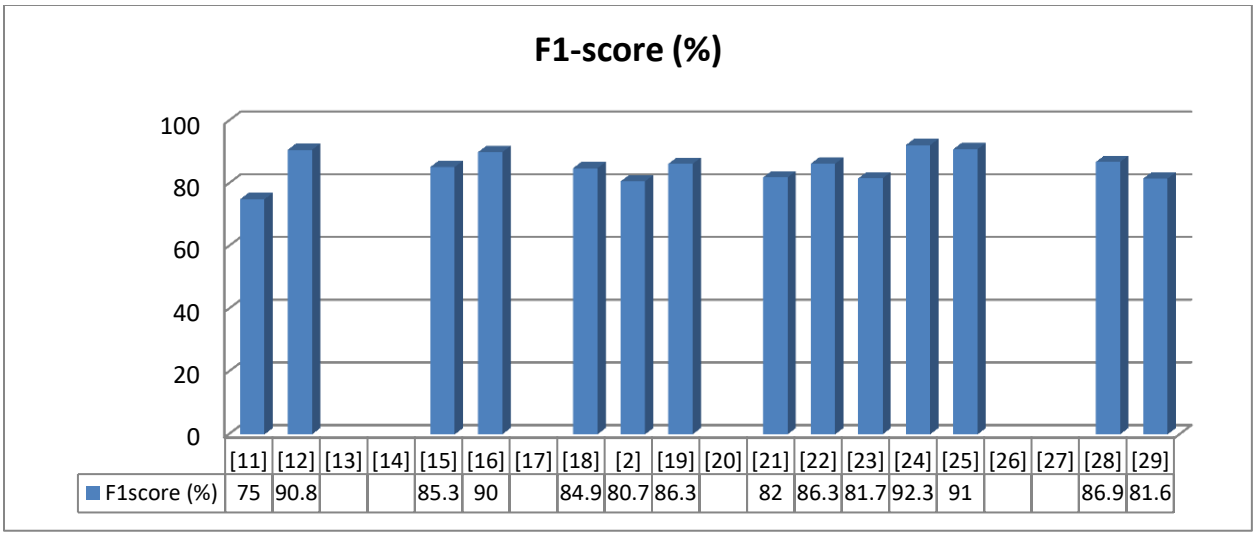

Figure 5. F1-score of multiple IDSs on NSL-KDD

6. Conclusions and Future Work

IoT is a great platform for joining users globally with no human involvement. These networks experience various modes of attacks and irregularities due to the lack of sensors supervising. To protect IoT systems, many IDSs were proposed. The majority of these approaches have limited scalability and accuracy. existing IDSs still face challenges in improving the detection accuracy, reducing the false alarm rate and detecting unknown attacks. This research aims to conduct a comparative study of intrusion detection systems applied to NSL-KDD dataset for two reasons. First, it's the most widely used dataset. Second, it is considered the key-stone in cyber security research field and most commonly used dataset in the field of cyber security [40]. This comparative study will pave the way for researchers for deep understanding the up-to-date improvement of IDSs that lead to better intrusion detection performance. It proves that combining supervised and unsupervised learning algorithms can significantly increase the accuracy of intrusion detection system, also choosing only features that have higher impact can do the same for accuracy. Feature selection algorithms show their ability to improve IDS accuracy with noticeable values whereas using all 41 features leads to increase in time and exactly improving IDS accuracy.

As a future work, we aim to conduct intensive research that focuses on improving the accuracy of IDS. In addition, we would like to investigate newly proposed datasets (e.g., CIC-IDS2017). Also, we can extend our work to intrusion prevention to prevent any intrusion which attends to harm the system.

References

[1] Morgan, S. "Cybercrime Magazine," Cyberwarfare In The C-Suite. 2020. [Online]. Available: https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021/

[2] Vinayakumar R., Alazab M., Soman K.P., Poornachandran P., Al-Nemrat A., and Venkatraman S., "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[3] Tavallaee M., Bagheri E., Lu W., and Ghorbani A.A., "A detailed analysis of the KDD CUP 99 data set," IEEE Symp. Comput. Intell. Secur. Def. Appl., pp. 1–6, 2009. doi: 10.1109/CISDA.2009.5356528.

[4] Hajisalem V., and Babaie S., "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," Comput. Networks, vol. 136, pp. 37–50, 2018. doi: 10.1016/j.comnet.2018.02.028.

[5] Liu H., and Lang B., "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," Appl. Sci., vol. 9, p. 4396, 2019. doi: 10.3390/app9204396.

[6] Hindy H. et al., "A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems," IEEE Access, vol. 8, pp. 104650–104675, 2020. doi: 10.1109/ACCESS.2020.3000179.

[7] Saranya T., Sridevi S., Deisy C., Chung T.D., and Khan M.K.A., "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," Procedia Comput. Sci., vol. 171, pp. 1251–1260, 2020. doi: 10.1016/j.procs.2020.04.133.

[8] Rashid A., Siddique M. J., and Ahmed S. M., "Machine and Deep Learning Based Comparative Analysis Using Hybrid Approaches for Intrusion Detection System," 3rd International Conference on Advancements in Computational Sciences (ICACS). pp. 1–9, 2020. doi: 10.1109/ICACS47775.2020.9055946.

[9] Malik R., Singh Y., Sheikh Z. A., Anand P., Singh P. K., and Workneh T. C., "An Improved Deep Belief Network IDS on IoT-Based Network for Traffic Systems," J. Adv. Transp., vol. 2022, p. 7892130, 2022. doi: 10.1155/2022/7892130.

[10] Meena G., Babita, and Mohbey K. K., “Assessment of Network Intrusion Detection System Based on Shallow and Deep Learning Approaches BT - Emerging Technologies in Computer Engineering: Cognitive Computing and Intelligent IoT,” , pp. 310–335, 2022.

[11] Tang T. A., Mhamdi L., McLernon D., Zaidi S. A. R., and Ghogho M., “Deep learning approach for Network Intrusion Detection in Software Defined Networking,” International Conference on Wireless Networks and Mobile Communications (WINCOM). pp. 258–263, 2016. doi: 10.1109/WINCOM.2016.7777224.

[12] Shone N., Ngoc T. N., Phai V. D., and Shi Q., “A Deep Learning Approach to Network Intrusion Detection,” IEEE Trans. Emerg. Top. Comput. Intell., vol. 2, no. 1, pp. 41–50, 2018, doi: 10.1109/TETCI.2017.2772792.

[13] Wu K., Chen Z., and Li W., “A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks,” IEEE Access, vol. 6, pp. 50850–50859, 2018, doi: 10.1109/ACCESS.2018.2868993.

[14] Ahmim A., Derdour M., and Ferrag M. A., “An intrusion detection system based on combining probability predictions of a tree of classifiers,” Int. J. Commun. Syst., vol. 31, no. 9, 2018, doi: 10.1002/dac.3547.

[15] Al-Qatf M., Lasheng Y., Al-Habib M., and Al-Sabahi K., “Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection,” IEEE Access, vol. 6, pp. 52843–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.

[16] Tang T.A., Mhamdi L., McLernon D., Zaidi S.A.R., and Ghogho M., “Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks,” in 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), 2018, pp. 202–206. doi: 10.1109/NETSOFT.2018.8460090.

[17] Kasongo S.M. and Sun Y., “A Deep Learning Method with Filter Based Feature Engineering for Wireless Intrusion Detection System,” IEEE Access, vol. 7, pp. 38597–38607, 2019, doi: 10.1109/ACCESS.2019.2905633.

[18] Gao X., Shan C., Hu C., and Liu Z., “An Adaptive Ensemble Machine Learning Model for Intrusion Detection,” IEEE Access, vol. 7, pp. 82512–82521, 2019, doi: 10.1109/ACCESS.2019.2923640.

[19] Yang Y., Zheng K., Wu C., and Yang Y., “Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network,” Sensors, vol. 19, no. 11, 2019, doi: 10.3390/s19112528.

[20] Papamartzivanos D., Gómez Mármol F., and Kambourakis G., “Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems,” IEEE Access, vol. 7, pp. 13546–13560, 2019, doi: 10.1109/ACCESS.2019.2893871.

[21] Ieracitano C., Adeel A., Morabito F.C., and Hussain A., “A novel statistical analysis and autoencoder driven intelligent intrusion detection approach,” Neurocomputing, vol. 387, pp. 51–62, 2020, doi: 10.1016/j.neucom.2019.11.016.

[22] Haggag M., Tantawy M.M., and El-Soudani M.M.S., “Implementing a Deep Learning Model for Intrusion Detection on Apache Spark Platform,” IEEE Access, vol. 8, pp. 163660–163672, 2020, doi: 10.1109/ACCESS.2020.3019931.

[23] Liu L., Wang P., Lin J., and Liu L., “Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning,” IEEE Access, vol. 9, pp. 7550–7563, 2021, doi: 10.1109/ACCESS.2020.3048198.

[24] Xu W., Jang-Jaccard J., Singh A., Wei Y., and Sabrina F., “Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset,” IEEE Access, vol. 9, pp. 140136–140146, 2021, doi: 10.1109/ACCESS.2021.3116612.

[25] Mashuqur A.K.M., Mazumder R., Kamruzzaman N.M., Akter N., Arbe N., and Rahman M.M., “Network Intrusion Detection Using Hybrid Machine Learning Model,” in International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 2021, pp. 1–8. doi: 10.1109/ICAECT49130.2021.9392483.

[26] Alotaibi S.D. et al., “Deep Neural Network-Based Intrusion Detection System through PCA,” Math. Probl. Eng., vol. 2022, p. 6488571, 2022, doi: 10.1155/2022/6488571.

[27] Wu T., Fan H., Zhu H., You C., Zhou H., and Huang X., “Intrusion detection system combined enhanced random forest with SMOTE algorithm,” EURASIP J. Adv. Signal Process., vol. 2022, no. 1, p. 39, 2022, doi: 10.1186/s13634-022-00871-6.

[28] Cui J., Zong L., Xie J., and Tang M., “A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data,” Appl. Intell., 2022, doi: 10.1007/s10489-022-03361-2.

[29] Thirimanne S. P., Jayawardana L., Yasakethu L., Liyanaarachchi P., and Hewage C., “Deep Neural Network Based Real-Time Intrusion Detection System,” SN Comput. Sci., vol. 3, no. 2, p. 145, 2022, doi: 10.1007/s42979-022-01031-1.

[30] “KDD Cup 1999.” [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup98/kddcup98.html

[31] Amiri F., Yousefi M.M.R., Lucas C., Shakery A., and Yazdani N., “Mutual information-based feature selection for intrusion detection systems,” J. Netw. Comput. Appl., vol. 34, no. 4, pp. 1184–1199, 2011, doi: 10.1016/j.jnca.2011.01.002.

[32] Xin Y. et al., “Machine Learning and Deep Learning Methods for Cybersecurity,” IEEE Access, vol. 6, pp. 35365–35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

[33] Karatas G., Demir O., and Sahingoz O.K., “Deep Learning in Intrusion Detection Systems," 2018 International Congress on Big Data,” in Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), 2018, pp. 113–116. doi: 10.1109/IBIGDELFT.2018.8625278.

[34] Xie J. et al., “A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges,” IEEE Commun. Surv. Tutorials, vol. 21, no. 1, pp. 393–430, 2019, doi: 10.1109/COMST.2018.2866942.

[35] Aledhari M., Razzak R., and Parizi R.M., “Machine learning for network application security: Empirical evaluation and optimization,” Comput. Electr. Eng., vol. 91, 2021, doi: 10.1016/j.compeleceng.2021.107052.

[36] S. Singh and S. Banerjee, “International Conference on Communication and Signal Processing (ICCSP),” IEEE International Conference on Communication and Signal Processing. pp. 976–980, 2020. doi: 10.1109/ICCSP48568.2020.9182197.

[37] Belgrana F.Z., Benamrane N., Hamaida M.A., Chaabani A.M., and Taleb-Ahmed A., “Network Intrusion Detection System Using Neural Network and Condensed Nearest Neighbors with Selection of NSL-KDD Influencing Features,” in 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2021, pp. 23–29. doi: 10.1109/IoTaIS50849.2021.9359689.

[38] Uğurlu M. and Doğru İ.A., “A Survey on Deep Learning Based Intrusion Detection System,” in 4th International Conference on Computer Science and Engineering (UBMK), 2019, pp. 223–228. doi: 10.1109/UBMK.2019.8907206.

[39] Lansky J. et al., “A Survey of Deep Learning Techniques for Misuse-Based Intrusion Detection Systems,” Res. Sq., 2021, doi: 10.21203/rs.3.rs-208981/v1.

[40] Maseno E.M., Wang Z., and Xing H., “A Systematic Review on Hybrid Intrusion Detection System,” Secur. Commun. Networks, vol. 2022, p. 9663052, 2022, doi: 10.1155/2022/9663052.