Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM
e-ISSN: 2788-5879 p-ISSN: 2788-5887

A comparison of several intrusion detection methods using the NSL-KDD dataset

Hazem Salim Abdullah Abdullah [1]*

[1] Directorate of Municipalities Nineveh Governorate, Mosul, IRAQ

*Corresponding Author: Hazem Salim Abdullah Abdullah

DOI: https://doi.org/10.31185/wjcms.251
Received 05 June 2024; Accepted 27 June 2024; Available online 30 June 2024

ABSTRACT: The increasing significance of cybersecurity underscores the critical necessity of addressing evolving methods of hackers. This research investigates the way to classify and predict cyber-attacks on the NSL-KDD dataset using intrusion detection methods the investigation contrasts the capabilities of various algorithms, including RNN, MLP, CNN-LSTM, and ANN, in recognizing attacks. The results indicate that both MLP and RNN have the greatest efficiency and effectiveness for different time frames. these findings demonstrate the necessity of Constant evaluation and enhancement of intrusion detection systems in order to remain aware of the dynamic nature of the cyber threat landscape. Addressing cybersecurity issues necessitates a comprehensive approach that combines computational enhancements, human talent, organizational policies, and regulatory frameworks in order to create a powerful and stable cybersecurity system.

Keywords: Cyber Security, intrusion detection system, Deep Learning, Machine learning

1. INTRODUCTION

Today, cybersecurity is considered an essential technology, as cyberattacks are increasing in frequency. Several different security protocols must be employed to safeguard assets from cyber intrusion... It is estimated that cybercrimes will result in a global economic loss of up to six trillion dollars by the end of 2021 [1]. This represents a significant financial impact on both the economy and the technology sector. The intrusion detection system (IDS) proves its efficacy in thwarting intrusion attempts and serves as an important security component that cannot be overlooked. Firewalls and antivirus software have limited effectiveness against various network threats. It is important to detect cyberattacks and collect extensive data about them when they occur. These two essential and necessary advantages are implemented to protect the infrastructure of information and communications technology (ICT). The use of Intrusion Detection Systems (IDS) offers inherent benefits that stem from its operational mechanisms. Intrusion Detection Systems (IDSs) are classified according to their detection algorithms, which are used for identifying anomalies and detecting abusive behavior [2]. Misuse detection relies on the presence of the security attack signature. In addition, due to the unavailability of its signature to Intrusion Detection Systems (IDSs), it is incapable of detecting new attacks. However, its distinctive characteristics enable accurate identification of familiar attacks. Anomaly detection systems in IDS use the attributes of network traffic to detect new attacks. To obtain optimal accuracy, it is necessary to train an anomaly detection system to differentiate between normal and aberrant behavior. In order to successfully finish this learning stage, a dataset pertaining to security is necessary. If the training of the anomaly detection system is completed properly, it has the ability to identify and predict both new and zero-day threats. This is an opportunity for ICT to develop appropriate measures to protect its resources.

1.1 Problem Statement

As previously said, researchers from all around the globe are working to build intrusion detection systems that may mitigate the issue of rising assaults. Here, we attempt to do a comparative analysis of the most recent anomaly IDS research conducted over the last three years, using machine learning and deep learning approaches on the NSL-KDD dataset. Since the NSL-KDD dataset is the most well-known cyber security benchmark, working on it is our main goal.

1.2 Contribution of Paper

This report presents a comparative analysis of many research publications utilizing the Network Security Laboratory Knowledge Discovery and Data Mining (NSL-KDD) Dataset that are linked to anomaly detection [3]. This is how the rest of the paper is structured. Background information and related work are provided in Section 2. Section 3 contains the Subjects and Methods. A detailed presentation of the NSL-KDD-based intrusion detection systems is given in Section 4. The models' implementation, assessment metrics, findings, and discussion are given in Section 5. Section 6 offers a conclusion and recommendations for further study at the end of this essay.

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 systems are used to identify and accurately classify attacks and intrusion attempts occurring on a host or network. Host-based intrusion detection systems (HIDS) are specifically designed to monitor and identify intrusion attempts on a single host, while network intrusion detection systems (NIDS) are designed to monitor and detect intrusion attempts on an entire network [4]. The aforementioned classification is based on the data source [5][6]. In addition, intrusion detection systems (IDS) can be divided into two types based on their operation method: active IDS (modifying the environment) and passive IDS (logging and alerting on unauthorized access) [7]. IDS can be divided into three subcategories: stateful protocol analysis, anomaly-based, and signature-based IDS. The last knowledge-based approach is called stateful protocol analysis, which is a specification-based approach, sometimes also called signature-based. It can successfully defend against known attacks, detect unknown attacks using anomaly-based detection, and detect unknown attacks using stateful protocol analysis [8]. In order to choose the right solution for each specific scenario, it is important to understand the goal of using an intrusion detection system (IDS).

2.2 Related work

A research study presented in [9] introduced a deep learning detection system using Deep Neural Network (DNN) for software-defined networks. This system only utilizes six fundamental network features (duration, protocol_type, src_bytes, dst_bytes, count, and srv_count) from the NSL-KDD dataset. The system achieved an accuracy of 75.75% for anomaly detection. The paper suggested the combination of Non-Symmetric Deep Auto-Encoder (NDAE) with Random Forest (RF) in order to increase the accuracy of current approaches for abnormality detection. The new method achieved an accuracy of 89.22% using the 13-class NSL-KDD dataset, which was the most increased reported accuracy until 2018. Additionally, the proposed method also resulted in a time saving of 98.81%. In research, the vector format raw traffic is converted into picture data format. Subsequently, the authors used a Convolutional Neural Network (CNN) intrusion detection model to enhance the accuracy, surpassing the performance of current Machine Learning based methods, achieving a precision of 79.48%. The study conducted in [12] introduced a method called Hierarchical Combining of Predictions of a Tree of Classifiers (HCPTC-IDS). The performance of this method was then compared to other approaches such as NB, FL, RIPPER, DT, ANN, and SVM. The model efficiently processes each record in 373 microseconds, demonstrating its rapid data traffic processing capabilities on the NSL-KDD Dataset, achieving an accuracy of 89.75%. The technique described in [13] combines Sparse Auto-Encoder (SAE) with Support Vector Machine (SVM). The Self-Taught Learning Intrusion Detection System (STLIDS) Utilizes Self-Taught Learning (STL) for data model and Support Vector Machines (SVM) for classification. This deep-learning intrusion detection system has an accuracy of 84.96%. The GRU-RNN, as introduced in reference [14], improves the efficiency of anomaly detection. They employed a Feedforward Deep Neural Network (FFDNN) for classification. Furthermore, studies have shown that the precision of the intrusion detection system is strongly influenced by the number of neurons used in the FFDDN classifier. This technique increases the precision to 87.74% by utilizing 30 nodes and 3 concealed layers. The study in [16] introduced a flexible ensemble learning framework that utilized the NSL-KDD dataset and included methods such as random forest, decision tree, and Deep Neural Network (DNN) to train the model. The proposed adaptive ensemble learning model is expected to have an accuracy rate of 85.2%. The paper [2] introduces the Scalable Hybrid Intrusion Detection Alertnet (SHIA) architecture, which examines network-level and host-level data to detect intrusions. In addition, it proposed utilizing a Deep Neural Network (DNN) to identify and detect cyber-attacks. The experimentation was conducted using NSL-KDD. The binary class classification achieved an accuracy of 80.1% using a one-layer deep neural network (DNN), while the multi-class classification achieved an accuracy of 78.5% using a five-layer DNN. The paper referenced as [17] presents the Improved Conditional Variational Auto-Encoder Deep Neural Network (ICVAE-DNN) model. The NSL-KDD dataset is used to evaluate this model. The accuracy, detection rate, and incorrect positive rate of the ICVAE-DNN model are then compared to six other classification algorithms: K-Nearest Neighbors (KNN), Multinomial Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), Deep Neural Network (DNN), and Deep Belief Network (DBN). It has been shown that its precision surpasses theirs. The accuracy achieved by

(ICVAEDNN) is 85.97%. Self-adaptive and autonomous misuse the IDS (Intrusion Detection System) was introduced in [18] and relies on Self-taught learning combined with a technique called MAPE-K. Self-taught learning is an advanced method of deep learning that can effectively detect previously unknown assaults by reconstructing unlabeled data. When combined with the MAPE-K reference model, it achieves an impressive accuracy of 77.99% in identifying these unseen attacks. The Auto-Encoder (AE) and statistical analysis model are introduced, using the NSL-KDD dataset from reference [19]. The results demonstrate the influence of adding or reducing Improving accuracy through hidden layers. Specifically, a single Hidden Layer (HL) consisting of 50 units achieves an accuracy of 84.24% for Binary classification and 87% for multi-Classification. A Deep learning system was introduced in [20] that demonstrated superior accuracy compared to previously existing systems, using a Spark Cluster setup. The system under consideration is referred to as DLS-IDS (Deep Learning Spark-IDS). The use of Long-Short Term Memory (LSTM) in conjunction with Synthetic Minority Over-Sampling Technique (SMOTE) resulted in a notable enhancement in the detection accuracy, achieving a commendable rate of 83.57%. The Difficult Set Sampling Technique (DSSTE) method [21] is proposed to enhance the classification model for learning unbalanced dataset data. The NSL-KDD dataset is utilized as a benchmark dataset. Their accuracy reached 82.84%. The study [22] shown a clear relationship between the accuracy of detection and the quality of data collection. To address this, the researchers suggested a 5-layer Auto Encoder model that improves the identification of anomalous network traffic with improved accuracy. The suggested methodology was evaluated using the NSL-KDD dataset, achieving a superior accuracy rate of 90.61%. In [23], a hybrid machine learning model and a feature selection strategy were proposed and deployed to the NSL-KDD dataset. The model picked seventeen characteristics for analysis. The suggested model achieved an accuracy result of 90.41%, surpassing other learning models by 11% in terms of accuracy and detection rate. The use of the ReLU activation function in deep neural networks, along with principal component analysis (PCA) as described in [24], results in a data processing acceleration and achieves an accuracy of 88.64%. The system proposed in [25], which combines an upgraded random forest with the synthetic minority oversampling technique (SMOTE), achieves an accuracy of 78.47%. The proposed system, GMM-WGAN, is a multi-module integrated intrusion detection system. It consists of three components: feature selection, imbalance processing, and classification. In a previous study [26], the system attained an accuracy of 86.59%. The proposed model is a deep neural network (DNN) that has been trained using 28 features from the NSL-KDD dataset. Additionally, a feature scaling technique described in [27] has been used. The model achieves an accuracy of 81.87%.

3. NSL-KDD Dataset

The KDD99 dataset, which was generated in 1999, became one of the most extensively used study datasets in the field of cyber security [28]. After extensive investigation on KDD99, researchers have identified some drawbacks that need resolution, including redundancy and the excessive quantity of records in both the training and testing datasets. These issues pose challenges when working with the whole dataset in experiments. In order to address the aforementioned drawbacks, a more recent iteration called NSL-KDD was suggested in [3]. Since 2009, NSL-KDD has been recognized as the primary dataset for cyber security research. The NSL-KDD dataset comprises of two subsets: KDDTrain+ with 80% or 125,973 records and KDDTest+ with 20% or 22,544 records. Each record consists of 41 characteristics, which are divided into four distinct categories: Basic features, time-based Traffic features, connection-based Traffic features, and Content features [29]. There are twenty-one predicted label classes assigned to each record, representing both attack and normal records. Inside the cyber security field, every individual piece of data is regarded as a session, representing a link between two hosts inside a network. The probability distribution of KDDTrain+ differs from that of KDDTest+. The test dataset includes some assaults that are not included in the training data. The training dataset consists of 24 distinct kinds of assaults, whereas the testing dataset includes an additional 14 types of attacks that are not included in the training set. This is done to evaluate the classifier's capacity to recognize unknown attacks. NSL-KDD provides a novel approach that enhances the KDD99 dataset. For instance, the KDD99 dataset classifies probing as an attack, but the NSL-KDD dataset does not classify it as an attack until the number of rounds exceeds a certain threshold. Table 1 provides a comprehensive overview of 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

As seen in Figure 1, Deep Learning (DL) is classified as a subset of Machine Learning (ML) methods. Machine Learning comprises a broad array of approaches that allow computers to learn from data. On the other hand, Figure 2 shows that Deep Learning primarily refers to methods that rely on neural networks with several layers, often known as deep neural networks [30].

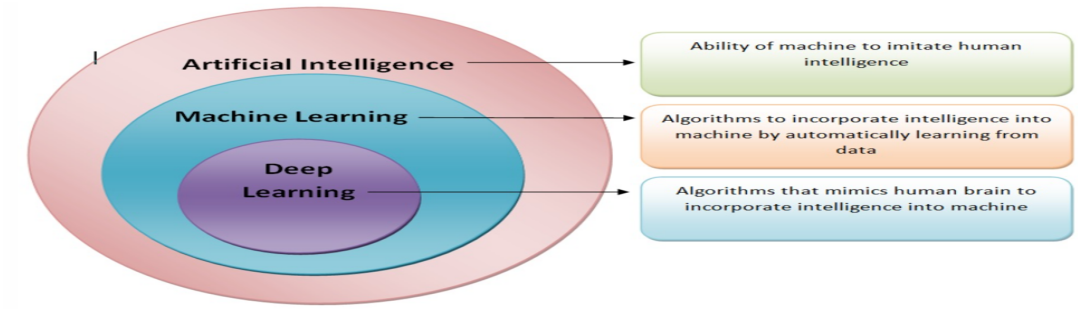

FIGURE 1. -A Venn diagram showing how DL is considered a subset of ML

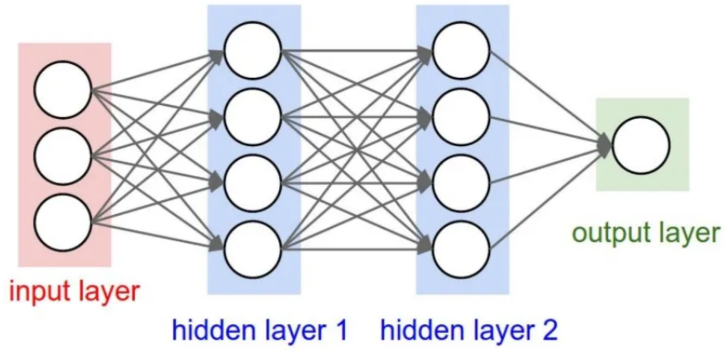

FIGURE 2. -Deep Learning Model

However, several writers argue that there are distinctions between DL and ML algorithms, notwithstanding the aforementioned factors. Table 2 outlines the primary distinctions between machine learning (ML) and deep learning (DL) [31]:

Table 2. - The main differences between ML and DL

#	ML	DL
1	It can use a small or medium amount of data to build good models for its algorithm, which works better with organized and structured data.	Thanks to its multi-layer architecture, it works and excels at processing large amounts of unstructured data, such as images, text, and documents.
2	It divides the problem into sub-problems and solves them one by one.	It solves the problem in one step using its multiple layers.
3	Its algorithms have a relatively simple structure, such as linear regression or decision tree.	It is based on a complex, multi-layered, interconnected artificial neural network that mimics the structure of the human brain.
4	It can run on a central processing unit (CPU).	It Requires a graphics processing unit (GPU) and more powerful hardware to function properly.
5	Its algorithms require greater human intervention to select, and process features to identify the correct input.	It can extract features automatically or with minimal intervention for the algorithm to learn and process from its errors and data.
6	In training, it takes a shorter time, But during testing, it becomes slow.	In training, it takes longer; However, during testing, it becomes faster.
7	Despite this, some of ML algorithms are still the fastest.	

In this section we will give a simple summary about the interested IDS algorithms as follows:

4.1 Artificial Neural Network (ANN)

Artificial Neural Network is a specialized branch of machine learning. The Artificial Neural Network (ANN) seeks to develop a machine learning system that is modeled after the biological structure of the human brain. Artificial neural

networks (ANNs) consist of many layers [32]. The structure consists of an input layer, a hidden layer, and an output layer. In the context of the input layer, every feature in the dataset is represented by a neuron. The input is sent to the subsequent layer. The Hidden Layer is comprised of a collection of neurons, with each neuron being given a weight. Each layer receives information from the preceding layer. The final outcome is derived from the output layer. Neural Networks need a significant amount of additional computational power. There are three sequential operations that need to be carried out in any neural network, as seen in Figure 3:

* Compute the predicted Y values (Ypred) by taking the input variables xi and utilizing the linear combination formula. Next, calculate the loss or error term. The error term is the difference between the observed values and the projected values Ypred.
* Increase the loss function or error term in order to reduce its importance.

4.2 The CNN-LSTM Model

The proposed model by the authors is comprised of Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM). The primary design of the Convolutional Neural Network (CNN) involves executing convolutional and pooling operations on the input data, then passing it through a fully connected layer that carries out the classification task [33]. The selected architecture, LSTM, is very consistent in translation and has a significant ability to predict the short term data sequence. In this study, the scientists chose to combine LSTM with CNN in order to recognize long-range dependencies in features. The authors employed the local nature of CNN to more comprehensively explore the data's properties. Later, they employed the sensitivity of the LSTM to the order of data characteristics in order to mitigate its effects. The The fusion approach attempts to take advantage of the diversity of learning methods to create a more powerful and accurate intrusion detection system. By combining methods like random forest, decision tree, and deep neural networks, the ensemble model can take into account different aspects of the data and have a more informed approach to potential cyber dangers.

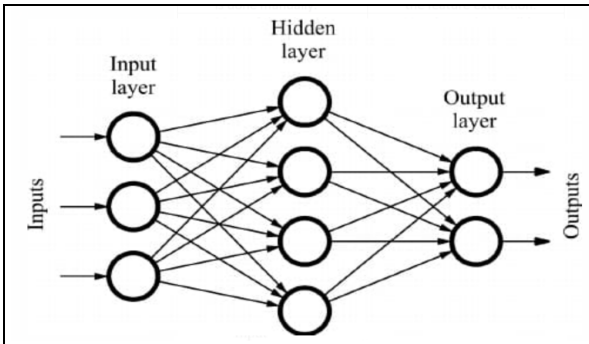

FIGURE 3. -Architecture of ANN

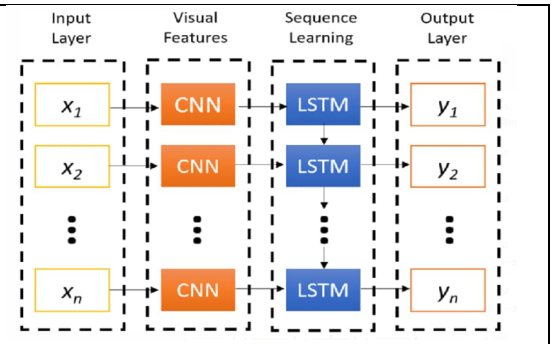

FIGURE 4. -Architecture of CNN-LSTM Model

4.3 The RECURRENT NEURAL NETWORK (RNN) Model

The RNN model is similar to the CNN model in terms of temporal structure. It's composed of three layers: input, a middle layer that has the ability to forward and backward propagate, and an output layer. RNNs have a superior capacity to deal with tabular data, as well as to classify, predict, and régress. The LSTM component of the RNN model helps solve problems associated with effectively predicting sequences. The structure of the RNN model is illustrated in Figure 5. [34] The RNN model has an input layer, an LSTM layer, hidden layers that are composed of a sigmoid function, and a final output layer that is dense. It employs the Adam optimizer and cross entropy loss.

4.4 MULTI-LAYER PERCEPTRON (MLP) Model

The Multilayer Perceptron model is sometimes also called the MLP model. It is a regression model that transforms input information into a complex structure. The structure of the data that is non-linear is inputted into the intermediate layer of the perceptron that is hidden, there, it is processed and then transmitted to the output layer. The middle hidden layer is composed of a Non-Linear function that performs regression predictions and addresses classification issues. The output is calculated by taking the weighted inputs and adding a bias to each layer's output. Figure 6 illustrates the Multilayer Perceptron (MLP) model.

The MLP model is often considered to be a universal emulator because of its underlying design, which is based on XOR operations. The MLP, or MultipleLayer Perceptron, is a type of neural network that employs the backpropagation algorithm for the purpose of learning. This algorithm is intended to specifically teach the model how to handle increasing functions. [35].

.

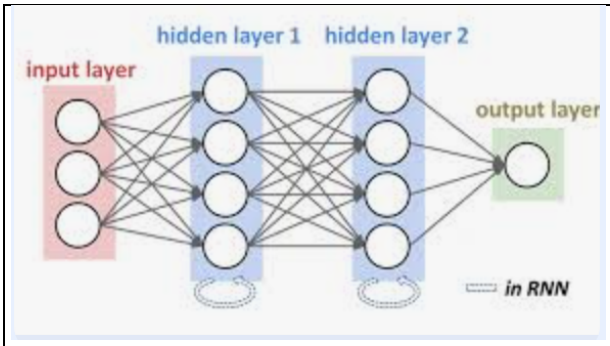

FIGURE 5. -Architecture of RNN Model

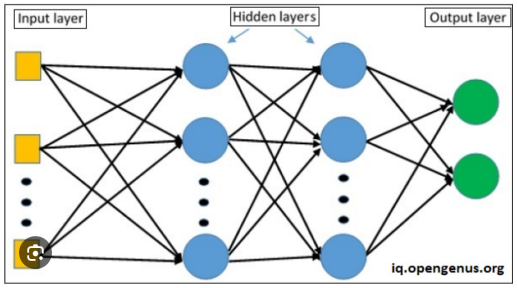

FIGURE 6. -Architecture of MLP Model

5. Models Implementation, Results and Discussion

5.1 Implementation of Proposed (RNN, MLP, CNNLSTM, ANN) models

- Utilize the MLP deep learning model as a reference to carry out the training process and optimize its parameters efficiently for intrusion detection with the NSL KDD dataset, according to: *Performing preprocessing*, normalization, and encoding on the NSL KDD dataset. *The model structure* is defined with three hidden layers, each consisting of 128 neurons, and utilizing the sigmoid activation function. *The model is compiled* using binary cross-entropy loss, Adam optimizer, and accuracy metric. *Performing training* for 10, 20, 40, and 100 epochs using a batch size of 32. *Optimizing* via grid search to determine the ideal learning rate (0.1) and batch size (32), Assess and verify the model's performance using a separate validation set, fine-tune the hyperparameters based on the accuracy achieved during validation. *Evaluating* the ultimate performance on the test set.
- So, to build all proposed Deep Neural Network models, it is essential to import the following libraries into the Google Collaboratory development environment: NumPy, Pandas, Scikit-learn, Keras, Tensorflow, Matplotlib, and Seaborn. The first phase of the experiments included importing, cleaning, and preparing 80% of data for the training process and 20% for testing data. The data preparation method for a deep neural network involves converting the target labels into binary form (normal or attack), encoding categorical data, transmitting the data as a NumPy array, and normalizing the data.
- Neural network training takes a lot of time, even with modern equipment. The training process will terminate before the model converges if the number of epochs is set too low during the model-building stage. However, if a very high number of epochs is selected, overfitting will happen. We'll waste time and computer resources, too.
- The notion of epochs is essential in the training process of machine learning models, namely neural networks. Choosing the appropriate number of epochs is essential to ensure that the model efficiently learns from the data without suffering from overfitting or underfitting. By monitoring the model's performance on a validation set and using strategies like as early halting, one may ascertain the ideal number of epochs.

5.2 Evaluation Metrics

- Accuracy is the percentage of correctly predicted incidents that are compared to the original (real) label. The more accurate the information, the more exact the created predictions become. Cybersecurity researchers attempt to improve the fidelity of their model to identify typical or unusual computer network incidents. Accuracy is determined by dividing the total number of accurate predictions by the total size of the dataset. The most accurate level is 1.

$$\text{Accuracy} = (TP + TN) / (TP + FP + TN + FN) \quad (1).$$

TP is the number of true positive cases, TN is the number of true negative cases, FP is the number of false positive cases, and FN is the number of false negative cases.

- Recall, also called sensitivity, is the percentage of violent incidents that are correctly classified. Recall is measured in terms of the fraction of accurate positive predictions among the total number of positive instances. The word "true positive rate" (TPR) is another term for the same. The optimal value for sensitivity is 1.

$$\text{Recall} = TP/(TP + FN) \quad (2).$$

- Precision is the percentage of correctly predicted attacks that are actually committed. The math involves taking the number of accurate positive predictions and dividing it by the total number of positive predictions. It's occasionally called the positive predictive value (PPV). The optimal degree of accuracy is 1.

$$\text{Precision} = TP / (TP + FP) \quad (3).$$

- A greater number for both recall and precision indicates better performance. The F1-Score metric is used to capture the advantages of recall and accuracy in a single measure. The harmonic mean of accuracy and recall is determined as follows:

$$\text{F1-Score} = 2TP / (2TP + FP + FN) \qquad (4).$$

- To address the problem of typical occurrences being recognized as attacks, the False Positive Rate may be used. It is calculated by dividing the count of false positive predictions by the total number of negatives. The optimal false positive rate is 0.0,

$$FPR = FP / (FP + TN) \qquad (5).$$

5.3 Results and Discussion

- The RNN, MLP, CNN_LSTM, and ANN algorithms were applied to several trials, considering the limits indicated earlier. The number of epochs used in each trial varied, with options ranging from 10 to 100 epochs.
- Table 3 presents a performance evaluation of the suggested Intrusion Detection Systems (IDSs). All of them have the similar requirements of utilizing the NSL-KDD dataset and evaluating accuracy using an appropriate number of epochs as a measure. Table 3 displays the precision of the suggested models on NSL-KDD. MLP and RNN have the best accuracy, whilst ANN exhibits the lowest accuracy.
- Table 4 presents a performance comparison of the IDSs indicated in section (2.2) in relation to the suggested IDSs. Each of the recommended Intrusion Detection Systems (IDSs) clearly demonstrates a greater accuracy value compared to the maximum value provided in section (2.2). The ANN IDS exhibits the lowest accuracy among the suggested IDSs, while its accuracy surpasses that of the 5LAE IDS mentioned in section (2.2).
- The training accuracy and validation accuracy curves obtained from each experiment provide an analysis of the experiment results. These curves play a crucial role in assessing the performance and behavior of a deep learning model. The MLP and RNN curves exhibit a similar pattern, diverging at the desired point between 10 and 17 epochs. However, the CNN_LSTM and ANN curves do not show a clear determination of their epochs. Based on the analysis of Figures No. 7, 8, 9, and 10, it can be concluded that 10 is the best number of epochs. As a crucial part of this analysis, the optimal number of epochs to prevent divergence between the two curves was selected as 10 epochs. Figure 10 shows the accuracy training and validation curve for this experiment.

TABLE 3. -Accuracy Values of Proposed Models on NSL-KDD With Respect to Epochs

MODEL	EPOCHS NO.			
	100	40	20	10
RNN	0.9980	0.9969	0.9939	0.9909
MLP	0.9963	0.9980	0.9947	0.9949
CNNLSTM	0.9952	0.9916	0.9909	0.9884
ANN	0.9685	0.9665	0.9620	0.9600

TABLE 4. - Performance comparison among IDSs on NSL-KDD dataset

Ref.	Approach	Acc.	Ref.	Approach	Acc.	Ref.	Approach	Acc.
[2]	DNN-SHIA	80.10	[16]	AENL	85.20	[24]	DT-PCADNN	88.60
[9]	DNN	75.80	[17]	ICVAE-DNN	86.00	[25]	ERF	78.50
[10]	NDAE	89.20	[18]	SMAPE-K	78.00	[26]	GMM-WGAN	86.60
[11]	CNN	79.50	[19]	AE	84.20	[27]	RT-IDS	81.90
[12]	HCPTC	89.80	[20]	DLS-IDS	83.60	Proposed IDSs	MLP	99.49 - 99.80
[13]	STL+SVM	85.00	[21]	DSSTE	82.80		RNN	99.09 - 99.80
[14]	GRU-RNN	89.00	[22]	5LAE	90.60		CNNLSTM	98.84 - 99.52
[15]	FEU-FFDNN	87.70	[23]	LGBM	90.40		ANN	96.00 - 96.85

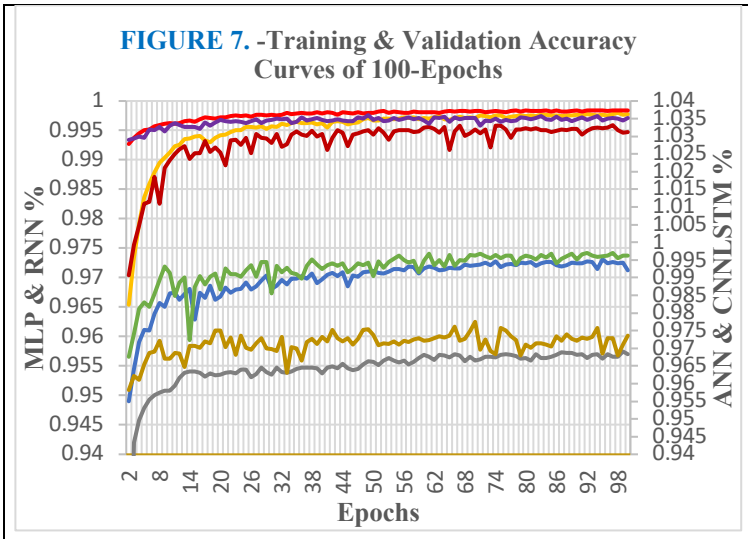

FIGURE 7. -Training & Validation Accuracy Curves of 100-Epochs

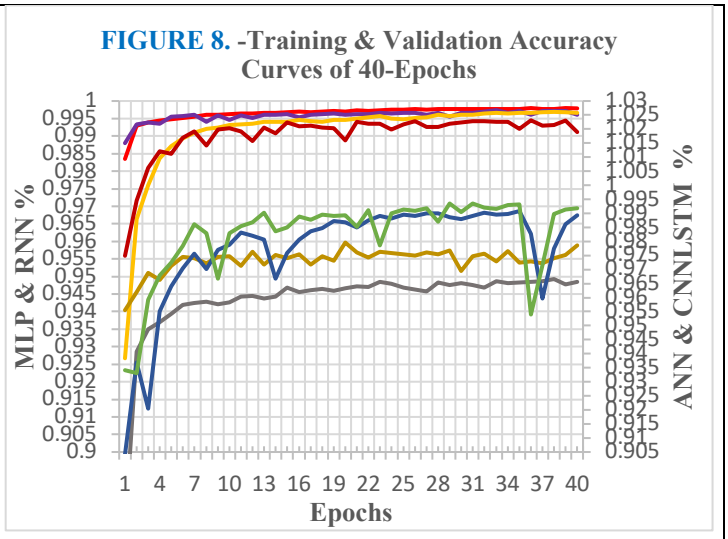

FIGURE 8. -Training & Validation Accuracy Curves of 40-Epochs

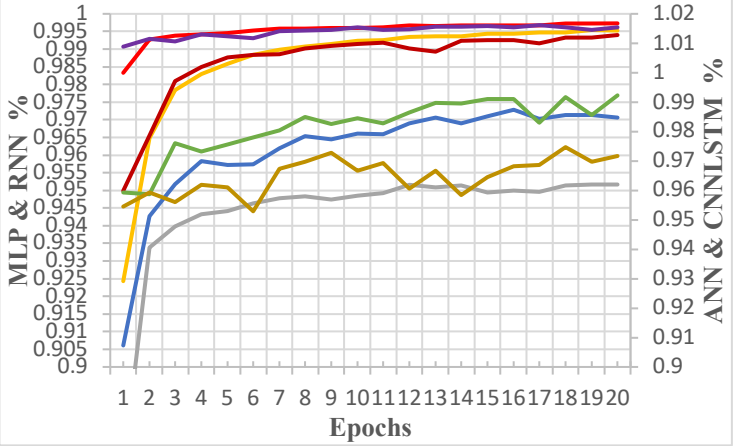

FIGURE 9. -Tranining & Validation Acuuracy Curves of 20-Epochs

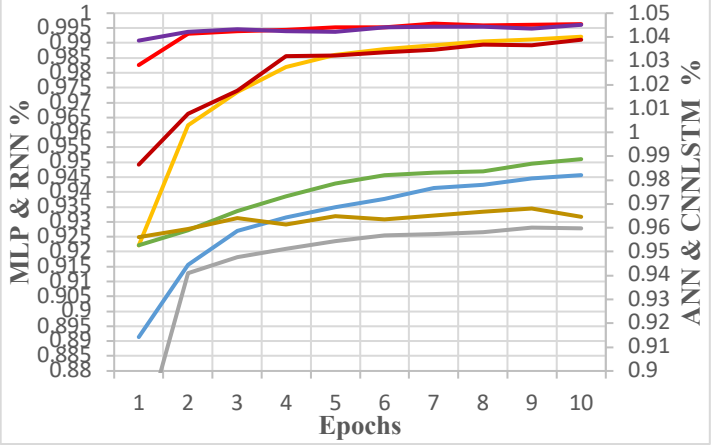

FIGURE 10. -Training & Validation Accuracy Curves of 10- Epochs

6. Conclusions and Future Work

- The Internet of Things (IoT) is an excellent platform for connecting consumers worldwide without the need for human intervention. These networks are susceptible to several types of assaults and abnormalities since they lack sensors for monitoring. Several Intrusion Detection Systems (IDSs) were suggested in order to safeguard IoT systems. Most of these methods have restricted scalability and precision. Current intrusion detection systems (IDSs) continue to encounter difficulties in enhancing detection accuracy, decreasing the proportion of false alarms, and identifying unknown assaults.
- The objective of this research is to do a comparative analysis of intrusion detection systems as they are applied to the NSL-KDD dataset. This work is driven by two specific goals. Firstly, it is the dataset that is most often used. Furthermore, it is widely regarded as the fundamental and most often used dataset in cyber security research [36].
- This comparison study will assist researchers in gaining a comprehensive grasp of the latest advancements in IDSs, which will ultimately enhance intrusion detection performance. The accuracy of the Table 3 indicates that the MLP and RNN models were more accurate than the CNN-LSTM and ANN models after 100 epochs. The combination of CNN and LSTM had a lower accuracy than the individual models, this is likely due to the higher complexity and the potential for overfitting. ANN had a simpler design than MLP, but it was less successful at recognizing intricate patterns. RNNs are more effective at sequentially processing data than basic ANNs.

- Future planned work will concentrate on enhancing the dependability of IDS through extensive study. Exploring the recently proposed datasets, such as the CIC-IDS2018, can help with this. Additionally, to prevent unauthorized access and threats to the system, a mechanism for intrusion prevention will be studied.

Funding

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] Morgan, S. "Cybercrime Magazine," Cyberwarfare in The C-Suite". 2020. [Online].vailable: 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," 1rd International Conference on Advancements in Computational Sciences (ICACS). pp. 1–9, 2020. doi: 10.1109/ICACS47775.2020.9055946.

[9] 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.

[10] 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.

[11] 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.

[12] 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. 13, no. 9, 2938, doi: 10.1002/dac.3547.

[13] 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. 52841–52856, 2018, doi: 10.1109/ACCESS.2018.2869577.

[14] 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.

[15] 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. 18597–38607, 2019, doi: 10.1109/ACCESS.2019.2905633.

[16] 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.

[17] 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. 39, no. 33, 2019, doi: 10.3390/s19112528.

[18] Papamartzivanos D., Gómez Mármol F., and Kambourakis G., "Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems," IEEE Access, vol. 7, pp. 31546–13560, 2019, doi: 10.1109/ACCESS.2019.2893871.

[19] Ieracitano C., Adeel A., Morabito F.C., and Hussain A., "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," Neurocomputing, vol. 187, pp. 53–62, 2020, doi: 10.1016/j.neucom.2019.11.016.
[20] 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. 361669–163672, 2020, doi: 10.1109/ACCESS.2020.3019931.
[21] 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. 7559–7563, 2021, doi: 10.1109/ACCESS.2020.3048198.
[22] 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.
[23] 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.
[24] Alotaibi S.D. et al., "Deep Neural Network-Based Intrusion Detection System through PCA," Math. Probl. Eng., vol. 2922, p. 6488571, 2022, doi: 10.1155/2022/6488571.
[25] 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. 2922, no. 3, p. 39, 2022, doi: 10.1186/s13634-022-00871-6.
[26] 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.
[27] 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. 1, no. 2, p. 145, 2022, doi: 10.1007/s42979-022-01031-1.
[28] "KDD Cup 3999." [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup98/kddcup98.html
[29] 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. 14, no. 4, pp. 3384–1199, 2011, doi: 10.1016/j.jnca.2011.01.002.
[30] Goodfellow, I., Bengio, Y., & Courville, A. (2016). "Deep Learning". MIT Press.
[31] Xin Y. et al., "Machine Learning and Deep Learning Methods for Cybersecurity," IEEE Access, vol. 6, pp. 15165–35381, 2018, doi: 10.1109/ACCESS.2018.2836950.
[32] Aledhari M., Razzak R., and Parizi R.M., "Machine learning for network application security: Empirical evaluation and optimization," Comput. Electr. Eng., vol. 93, 2923, doi: 10.1016/j.compeleceng.2021.107052.
[33] Jinhai, Song., Zhiyong, Zhang., B., Gupta. (2023). A Novel CNN-LSTM Fusion-Based Intrusion Detection Method for Industrial Internet. International Journal of Information Security and Privacy, doi: 10.4018/ijisp.325232
[34] B. Mohammed, E. K. Gbashi, (2021), "Intrusion Detection System for NSL-KDD dataset based on deep learning and recursive feature elimination," Engineering and Technology Journal, Vol. 39, No. 07, pp. 1069-1079, 2021. DOI: https://doi.org/10.30684/etj.v39i7.1695
[35] (2022). Improving Multilayer-Perceptron(MLP)-based Network Anomaly Detection with Birch Clustering on CICIDS-2017 Dataset. doi: 10.48550/arxiv.2208.09711
[36] 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