AN EFFICIENT DEEP LEARNING APPROACH FOR NETWORK INTRUSION DETECTION SYSTEM ON SOFTWARE DEFINED NETWORK

Mhmood Radhi Hadi and Adnan Saher Mohammed

Department of Computer Engineering, Karabük University, Karabük, Turkey

ABSTRACT

Software-defined Networking (SDN) is a new technology for changing network architecture and making it more flexible and controllable. SDN can control all tasks of a network through the controller. Providing security for controller consider extremely important. Due side of the controller on the network side Network intrusion detection system (NIDS) will be effective to provide security for the controller. In this study, we suggest building a system (NIDS-DL) to detect attacks using 5 deep learning classifiers (DNN, CNN, RNN, LSTM, GRU). Our approach depends on the binary classification of the attacks. We used the NSL-KDD dataset in our study to train our deep learning classifiers. We employed 12 features extracted from 41 features using the feature selection method. CNN classifiers harvest the highest results in most evaluation metrics. Other classifiers also achieved good results. We compared our deep learning classifiers with each other and with other related studies. Our approach achieved success in identifying the attacks and might be used with great efficiency in the future.

KEYWORDS

Network Intrusion Detection System, Software Defined Networking, Deep Learning, NIDS, SDN

1. INTRODUCTION

The architecture of traditional networks has not changed for decades to rum that it suffers from many problems and singled out security problems. Software-defined networking new solution or approach to address these problems, and it is characterized by many features that make it the future structure of the Internet. The most prominent feature of this network is that it is inexpensive, flexible, expandable, and increases the size of its infrastructure without the complexity of the traditional network. All operations in this architecture are controlled by a controller [1]. Instructions are exchanged between the controller and the switches via the OpenFlow protocol. The SDN architecture has many advantages, as it provided many solutions to the problems of the old network infrastructure, which made it the focus of attention and interest of authors [2]. OpenFlow protocol is based on the concept of different IP packets that are exchanged between the controller and the switches. SDN provided a comprehensive overview of the entire network through the controller controlling the entire network. The controller is considered the brain of the network, which is completely isolated from the network, and targeting it from attackers means the fall of the entire network. Accordingly, the controller is the most harmful part and the most affected by attacks. It is necessary to have a network intrusion detection system (NIDS) located in the network that protects the SDN, especially the controller that is in the network part from attacks, detecting and reducing their impact. There are several types of NIDS, an approach that uses a signature, that relies on data from previous attack logs that are stored and requires continuous updating, is called the signature-based NIDS approach [3], and a second approach that uses anomaly detection that monitors the traffic pattern is more

DOI: 10.5121/ijnsa.2022.14401

efficient and effective is called the NIDS approach Based on anomaly detection [4], which compares traffic behavior to normal and abnormal traffic. Machine learning is used with NIDS to identify attacks, but the efficiency is low. Within NIDS, a flow-based approach and anomaly detection are used together. Many factors have led to the lack of success and reliability of using machine learning in intrusion detection techniques in networks, the most prominent of which is the complexity to handle huge amounts of data that are unclassified where the performance and reliability of these systems are inefficient. Deep learning technology is a new and recent technology that predicts the possibility of solving machine learning problems, and it can deal with inconsistent data, find possible correlations, and give good and reliable performance. A reliable NIDS approach can be designed with accuracy and performance using deep learning. With deep learning, various attacks can be identified with high accuracy and with a high detection rate. SDN protection using NIDS based on deep learning is an effective method and a powerful defense mechanism. NIDS focuses on the detection of types of traffic as normal or abnormal behavior. Attacks cannot be completely prevented, but they can be detected early and identified, and their impact reduced if effective methods such as deep learning methods are used [5]. We propose a (NIDS-DL) approach for SDN using deep learning. More than one type of deep learning algorithm has been used to evaluate it based on several Metrics such as (Accuracy, F-score, Recall, Precision, etc.). we applied features selection methods to train our classifiers on high correlation features. The approach was applied to an NSL-KDD [6] dataset.

This paper is organized as follows: Section 1 Introduction. Section 2 is Related work that described some relevant previous work. Section 3 Proposed Methodology that clarified the proposed approach, also explains in brief classifiers model used and summary of architecture. Section 4 discussed the dataset and preprocessing methods applied. Section 5 Experiment results of the approach. Finally, Section 6 explains the conclusion and future work for the approach.

2. RELATED WORK

The application of machine learning systems with SDN has attracted the attention of many authors.

In [7] the author's purpose approach was based on five types of machine learning algorithms (RF, Naïve Bayes, SVM, CART, J84) to obtain an accurate and high-performance approach, this approach was applied to the NSL-KDD dataset with the employs 41 features, this approach achieved good detection accuracy in recognition of attacks and anomaly detection, the RF algorithm achieved the highest accuracy rate of 97%.

After the emergence of deep learning technology, several authors attempted to design several systems that use deep learning in NIDS for SDN in their approach. Authors in [8] the authors built a deep learning-based network intrusion detection approach for the SDN environment, using the DNN algorithm in their approach. Six features from the NSL-KDD dataset were used. The authors contrasted the outcomes of his approach with machine learning classifiers. The approach exhibited high detection accuracy and better performance than the machine learning classifier approach, demonstrating the feasibility and potential of using deep learning to construct network intrusion detection systems for SDN. the authors compared the results of the approach he used with machine learning classifiers.

Also, a study in [9] the same author proposed using a hybrid deep learning approach, the goal was to improve the accuracy and reach a better and more applicable approach, these approaches used two types of deep learning classifiers Gated recurrent unit and Recurrent Neural network to design a hybrid approach called (GRU-RNN), apply this approach was based on NSL-KDD dataset, where the author used in his approach six features in training the classifier. The hybrid

approach method achieved 89% better accuracy and proved to be superior to the previous method, as well as its easy and flexible application in the SDN working environment.

Another work in [10] The goal of this approach was to build intrusion detection systems for SDN, the researcher used machine learning and deep learning systems to compare the results. A deep learning algorithm (GRU) was used in the approach, the algorithm achieved better accuracy and performance than machine learning classifiers, more than one type of dataset was used in training and comparison, and six types of different attacks were categorized with a benign approach, the approach achieved great success indicating the possibility of applying deep learning in NIDS with great efficiency to SDN.

In this paper, several types of deep learning classifiers (CNN, DNN, RNN, LSTM, GRU) are applied. NSL-KDD dataset was used as the approach was applied to 12 features extracted. Each classifier was evaluated based on a different set of metrics. A broad approach to deep learning and its classifiers has been used to build a robust and effective NIDS system for detection and identifying attacks for future application within the SDN environment, which differs from the rest of the research in that it relies on more than metrics in assessment, not just accuracy and trying to get the best and highest result compared to related work.

3. PROPOSED METHODOLOGY

3.1. System Methodology Description

The adoption of most of the methods applied in the machine learning approach will become less effective with the development of attack and penetration systems and the tools used for them. The machine learning method needs more configured data and is also fewer data to process, moreover performance and accuracy become poor. Most of the methods that use deep learning, discussed by the authors, use classifiers. The classifier is mainly evaluated on the accuracy of the matching metric, and the accuracy is also low, which does not lead to building a reliable and efficient NIDS system to detect attacks.

All of these prompted us to build our methodology shown in Figure 1, this methodology is based on building the NIDS-DL approach for SDN, this approach uses more than one classifier for deep learning with training classifiers on 12 features extracted from 41 features in the NSL-KDD dataset, training the classifier on best correlation features will lead to the possibility of detecting various attacks. Applied feature selection method to select the best features that are effective and get correlations on the result, also the system will be powerful and reliable against attacks. The approach is evaluated on several Metrics and the classifiers are compared with each other.

In our approach, we evaluated CNN, DNN, RNN, LSTM, GRU classifiers are used, Results are compared where the (normalization) mechanism is used on the data to speed up the training process and get the best possible outcomes for generating an efficient NIDS classifier, also using feature selection method to avoid missing in training algorithm and try to reach the best accuracy and performance through selecting the best feature for training.

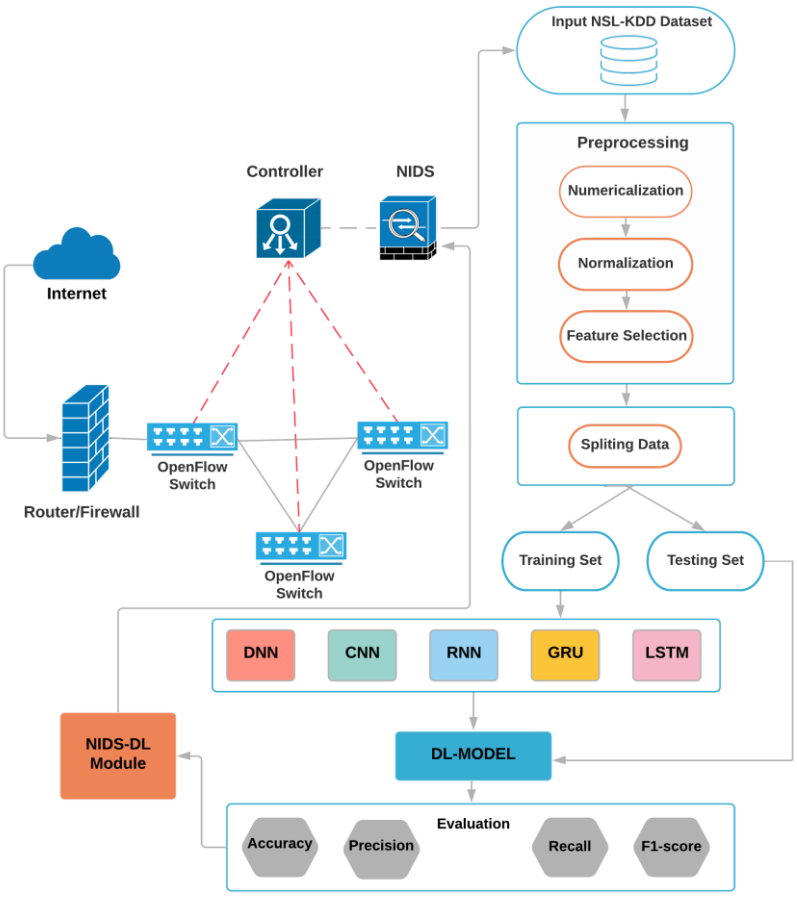

Figure 1. Proposed Methodology for (NIDS-DL) in SDN

3.2. Model Classifiers

In our study we use DNN, CNN, RNN, LSTM, GRU classifiers, architectures summary is given visualization in tables 1-5 we try to use little of layers in our classifier with try reach the best result and performance due to our approach will be applied in future and performance is important.

a) DNN is a deep neural network, an evolved class of simple neural networks. Table 1 shows the DNN architecture, the number of layers, and the types of tuning parameters used.

b) CNN is a kind of neural network that processes image data with high efficiency. Table 2 shows the internal architecture of the CNN classifier used in our approach we can see applying the RELU activation function inside the dense layer and also using 2 layers from MaxPooling to select a high value from each feature map.

c) RNN is Recurrent neural networks are also considered one of the simple neural networks, also considered a powerful type developed in the eighties. The most important thing that distinguishes this type and makes it a strong type is that it contains the internal memory we try in RNN used a little layer to train the classifier fast. Table 3 shows the architecture of RNN.

d) LSTM is Long short-term memory is one of the types of a type of RNN. It came to address several problems that the RNN suffers from. LSTM has the feature of retaining data and information stored for an extended period. Table 5 shows the architecture of LSTM.

e) GRU is Gated Recurrent Unit is also a type of standard recursive network. The specific architecture and interior design are similar to LSTM. Gated Recurrent Unit is designed to address the vanishing gradient problem in RNN. Table 4 shows the architecture of GRU.

Table 1. Summary of DNN model.

Variable	**Parameter**
Dense	3 Layers
Flatten	1 Layer
Dropout	2 Layers
Activation Function	Sigmoid
Optimizer Function	Adam
Loss Function	Binary Cross Entropy
Output	2
Epoch	100

Table 2. Summary of CNN model.

Variable	**Parameter**
Convolutional1D	4 Layers
Dense	2 Layer
Flatten	1 Layer
Dropout	1 Layers
MaxPooling	2 Layers
Activation Function	Sigmoid & RELU
Optimizer Function	Adam
Loss Function	Binary Cross Entropy
Output	2
Epoch	100

Table 3. Summary of RNN model.

Variable	**Parameter**
Simple RNN	2 Layers
Dense	1 Layer
Dropout	2 Layers
Activation Function	Sigmoid
Optimizer Function	Adam
Loss Function	Binary Cross Entropy
Output	2
Epoch	100

Table 4. Summary of GRU model.

Variable	Parameter
GRU	1 Layer
Dense	1 Layer
Dropout	1 Layer
Activation Function	Sigmoid
Optimizer Function	Adam
Loss Function	Binary Cross Entropy
Output	2
Epoch	100

Table 5. Summary of LSTM model.

Variable	Parameter
LSTM	1 Layer
Dense	1 Layer
Dropout	2 Layer
Activation Function	Sigmoid
Optimizer Function	Adam
Loss Function	Binary Cross Entropy
Output	2
Epoch	100

4. DATASET

In this part, we will discuss the NSL-KDD [11] dataset that was used in our proposed approach. The NSL-KDD dataset is an update and development of the KDDCup99 dataset [12], which is much older than it was proposed in 1999, as it contained several problems and contained null or it is a recursive dataset which many of its problems have been solved in the NSL-KDD dataset, but this does not mean that it does not contain mistakes. NSL-KDD contains 41 features, we extracted 12 features are more corrections using the feature selection method. NSL-KDD is used as a simulator for network data and internet traffic as it was used in several research and applied by authors in their approach. The main feature of the NSL-KDD dataset that made it preferable to many authors is that its size is almost consistent and contains reasonable several features that help in obtaining the best and most reliable classifiers.

4.1. Data Preprocessing

In this section, we will discuss the methods used in preprocessing datasets.

4.1.1. Numericalization

To handle the NSL-KDD dataset into deep learning classifiers, all data must be in numeric format. The NSL-KDD dataset contains three non-numeric features and 38 numeric features. The features are converted to numeric form so that they can be handled by classifiers after they are converted to array form. The features that are converted are ('flag','service','protocol_type'). For example, the feature ('protocol_type') contains three types of data ('icmp','udp','tcp'), which are encoded into (1,0,1), (1,1,0), (0,0,1). After using this method, all the 12 turns into a map of 122-dimensions.

4.1.2. Normalization

The normalization mechanism is applied for several tasks, it is used to speed up the training process for classifiers as it works to make the data set consistent and make the difference between the data small when we have the difference between the big and small data is large. Among the features in the NSL-KDD data set and contains the difference between its data are dst_bytes [0,9.11×109], duration [0,58329], src_bytes [0,9.11×109]. The formula shown in 1 is applied, which transforms the data range and makes it between [0,1].

$$xi = (xi - Min) / (Max - Min) \quad (1)$$

4.1.3. Feature Selection

We used data regularization by Pearson correlation coefficient consider significant statistical tool to select the best feature have a correlation more than the threshold value (0.5) with target value. After employing the feature selection method on NSL-KDD we extracted 9 from 38 numerical features that have a high correlation. Feature selection method we applied on numerical feature. Table 6. illustrates 12 features from NSL-KDD dataset will using inside classifiers.

Table 6. Feature extracted from NSL-KDD dataset.

No.	Features	No.	Features
1	protocol_type	7	srv_serror_rate
2	service	8	same_srv_rate
3	flag	9	dst_host_srv_count
4	count	10	dst_host_same_srv_rate
5	logged_in	11	dst_host_serror_rate
6	serror_rate	12	dst_host_srv_serror_rate

4.1.4. Data Splitting

The features are a selection from NSL-KDD Dataset are splitting by 75% for training and 25% for testing. Table 7. Showing partitioning of training and testing data into the NSL-KDD dataset with 12 features.

Table 7. A distribution instance of the NSL KDD dataset.

	Training set	Test set
Number of instances	94,479	31,494

4.2. Evaluation Metrics

To build a reliable (NIDS-DL) approach and use it efficiently, you need to achieve high results when evaluated according to various metrics, especially accuracy. We evaluated the approach using various metrics such as accuracy, recall, precision, and F1-score. All of these metrics are based on parameters specific to the confusion matrix (TP, TN, FP, FN). Table 7 The metrics used to evaluate our approach (NIDS-DL).

Table 8. Metrices evaluation of deep learning with formulas.

Metrics	**Formula**
Accuracy	TP + TN / TP +FP +TN + FN
Recall	TP / TP +FN
Precision	TP / TP + FP
F1-score	2 * P * R/ P + R
Receiver operating characteristic	TPR = TP / TP + FN

5. EXPERIMENT RESULTS

The goal of our approach is to try to get the best results for several metrics. The approach was made and implemented using the Python 3.5.6 programming language, also using (TensorFlow, Keras) with (NumPy, Pandas) library for preprocessing. The computer Hardware configuration is (Intel i7-2720 QM, 16 GB of RAM, AMD Radeon 2 GB, 256 GB SSD).

Comparing the results of the deep learning classifier with the accuracy metric shows that the CNN classifier achieves the highest results, followed by the DNN classifier, as shown in the Figure 2. When it comes to Precision metrics, CNN is the highest, followed by DNN. The RNN classifier harvest highest result in recall metrics, followed by CNN. Finally, the F1-score metrics that produced the highest result are the CNN classifier followed by the DNN, as shown in Figure 2 and Table 9.

Table 9. Evaluation Metrics Classifiers.

DL-Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
DNN	0.9853	0.983	0.9896	0.9863
CNN	0.9863	0.9845	0.9898	0.9872
RNN	0.9813	0.9751	0.9902	0.9826
LSTM	0.9804	0.9767	0.9856	0.9816
GRU	0.9778	0.973	0.9856	0.9793

Figure 2. Results evaluation metrics of deep learning classifiers.

The confusion matrix is also important when evaluating classifiers. The purpose of using the confusion matrix is to get the highest score in (TP) and (TN) and reduce the values in (FP) and (FN). From figure 3, we can see that the CNN classifier has the highest values in TP and the lowest in FP, and the RNN classifier has the highest values in (TN) and the lowest in (FN). Figure 3 shows the harvest results for all classifiers in confusion matrix metrics.

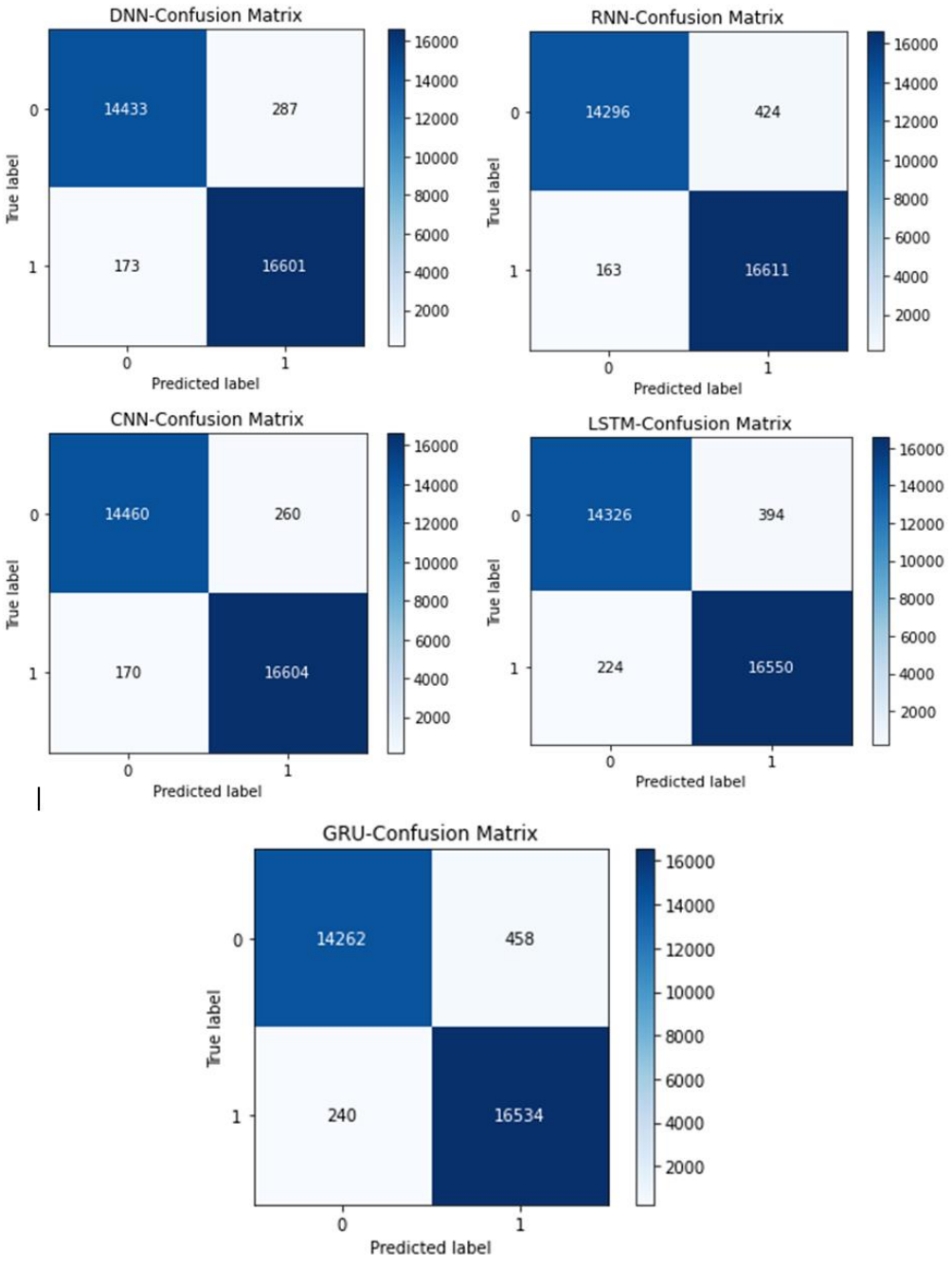

Figure 3. Results of Confusion Matrix for Deep learning classifiers.

ROC Curve is also an important metrics evaluation based on the sensitivity and specificity when measuring for each classifier. It can be seen that the CNN and DNN classifiers harvest similar results, and the LSTM, RNN classifiers also achievement similar results as shown in the Figure 4.

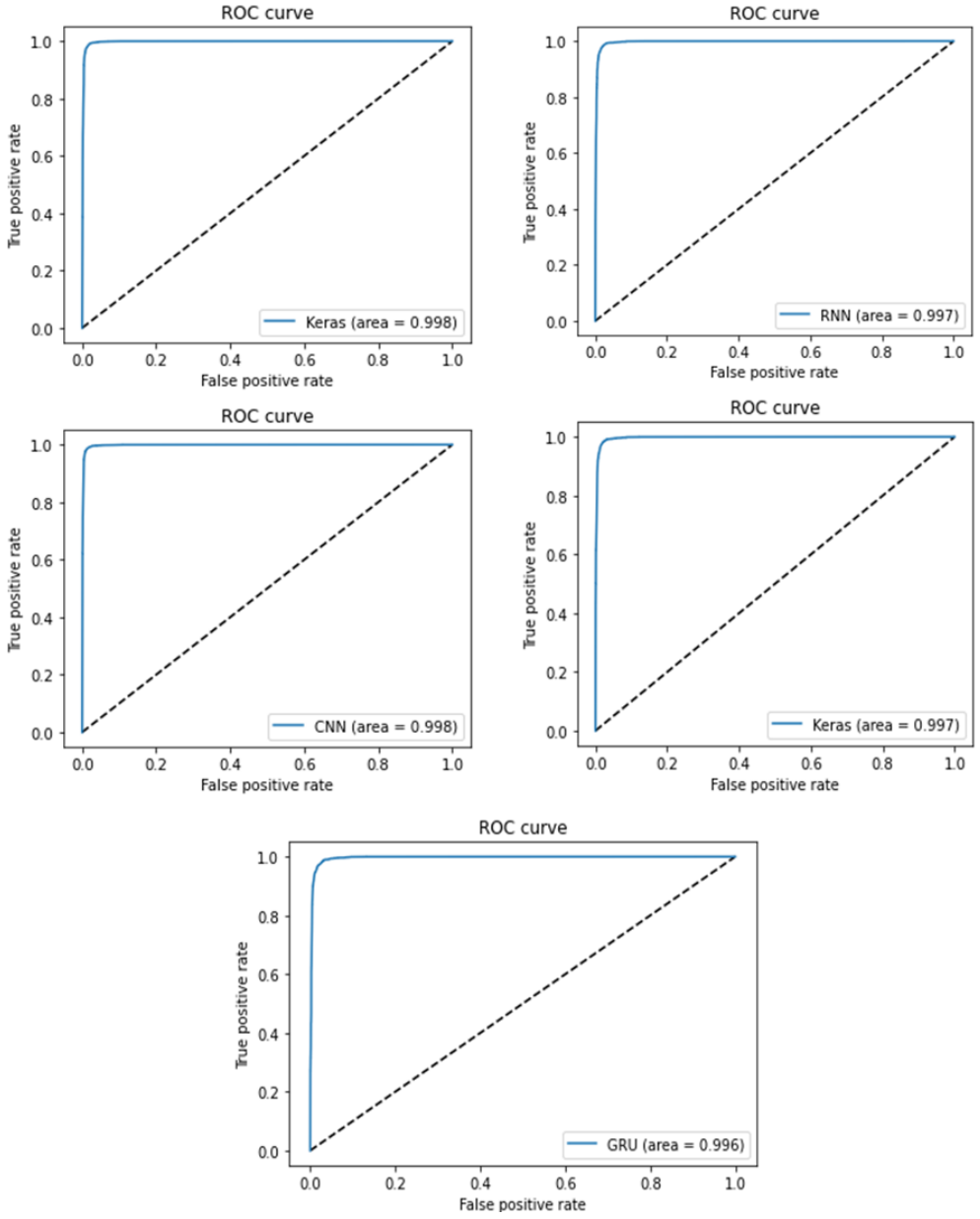

Figure 4. Results of ROC curve for deep learning classifiers.

We can show the overfitting for our classifier during the training through the curves of loss and accuracy with value of epoch. Aim in this curve is try to minimize value of loss and maximize of value of accuracy with adjusting between train and test. Harvest best accuracy for DNN classifier during training on the range epoch 93 as we see in figure 5. We can see from loss plot for DNN that need some enhancement for classifier due the wide difference between the train and test loss.

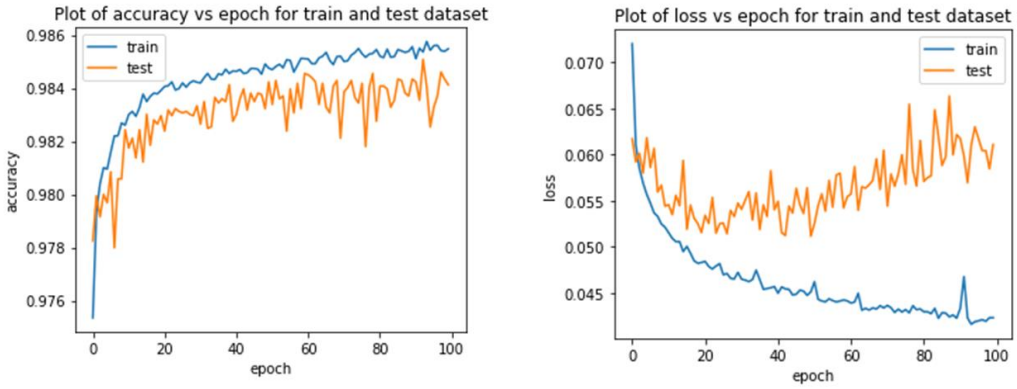

Figure 5. Performance of DNN classifier based on accuracy and loss curve.

CNN classifier is reaching the best accuracy in epoch 98 and best loss. As shown in figure 6. Also, we can see in loss plot that need some enhancement and adjusting for classifier due some wide between the training and testing that happen.

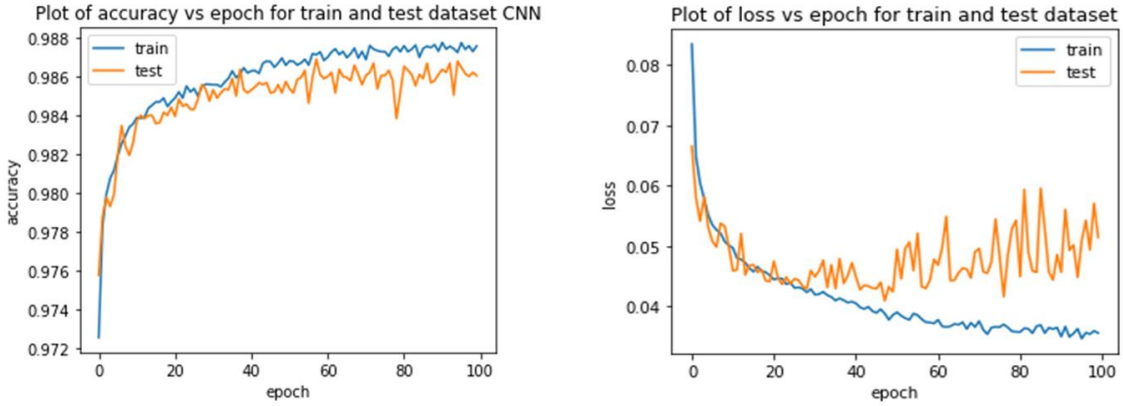

Figure 6. Performance of CNN classifier based on accuracy and loss curve.

The accuracy and loss plot for the RNN classifier as we can show in figure 7. As we see the classifier reaches the overfitting accuracy in 20 epochs with the same range of training and testing and for loss.

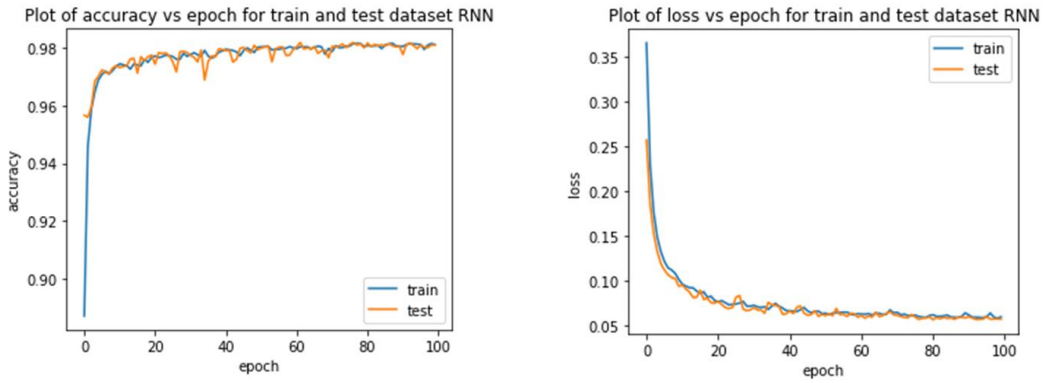

Figure 7. Performance of RNN classifier based on accuracy and loss curve.

LSTM classifier is reach to best accuracy and overfitting in 20 epochs as we can see in figure 8.

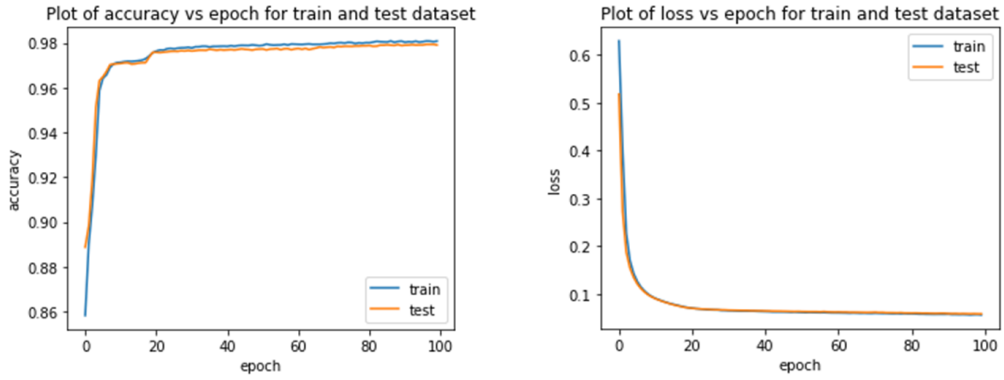

Figure 8. Performance of LSTM classifier based on accuracy and loss curve.

GRU classifier is reach to best accuracy and overfitting in 50 epochs as showing in figure 9.

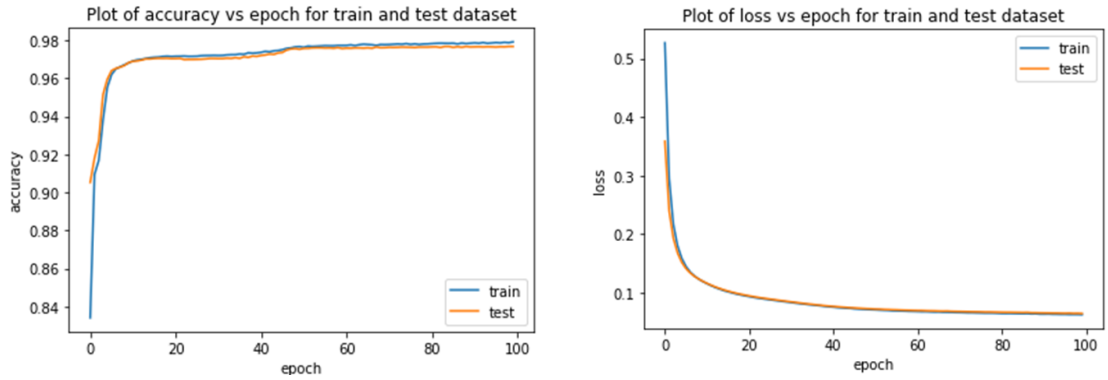

Figure 9. Performance of GRU classifier based on accuracy and loss curve.

We compared the results of our research with the results of other studies that used a similar to our approach, with a mention of the type of algorithm used and the evaluation on which each study relied as we can see in table 10.

Table 10. Accuracy Result Comparison with another Study Related.

Ref. Study	Method	Our approach results
[8]	DNN, ACC=75.75%	DNN, ACC=98.54%
[13]	RNN, ACC=90.53%	RNN, ACC=98.13%
[15]	LSTM, ACC=96.48%	LSTM, ACC=98.04%
[14]	CNN, ACC=96.43%	CNN, ACC=98.63%
[10]	GRU, Precision=89%	GRU, Precision=97.3%

6. CONCLUSION

In this paper, more than one type of deep learning algorithm is used and applied to detect abnormality in NIDS. The approach was evaluated on different metrics and the approach achieved high and reliable results. One of the most contributions of this work is using the feature selection method to train the classifiers on most feature correlations and avoid miss led during

training to reach the best result. Our approach focused on binary classification using deep learning algorithms. The results of the algorithms are compared with each other, the results of some classifiers are close, and the CNN classifier achieved the highest results. Also, we compare our classifiers with another related study and also get the highest result when compare. The use of deep learning demonstrated the possibility and superiority when applied in the binary classification of network intrusion detection systems. Since the proposed approach harvest high results, future work will be to evaluate the results of classifiers on more than one type of dataset and compare the results. A hybrid approach of deep learning algorithms can also be used as a future work, and its results compared with our approach. These approaches can also be used to detect a specific type of attack, such as (DOS) attacks also we apply this approach inside an SDN environment. Employed this study on different technology such as Blockchain or IoT and cloud.

REFERENCES

[1] "Software-Defined Networking (SDN) Definition - Open Networking Foundation." https://opennetworking.org/sdn-definition/ (accessed Apr. 25, 2022).

[2] McKeownNick *et al.*, "OpenFlow," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008, doi: 10.1145/1355734.1355746.

[3] S. Sangeetha, B. Gayathri Devi, R. Ramya, M. K. Dharani, and P. Sathya, "Signature Based Semantic Intrusion Detection System on Cloud," *Adv. Intell. Syst. Comput.*, vol. 339, pp. 657–666, 2015, doi: 10.1007/978-81-322-2250-7_66.

[4] S. K. Dey and M. M. Rahman, "Effects of Machine Learning Approach in Flow-Based Anomaly Detection on Software-Defined Networking," *Symmetry 2020, Vol. 12, Page 7*, vol. 12, no. 1, p. 7, Dec. 2019, doi: 10.3390/SYM12010007.

[5] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *Proc. - IEEE Symp. Secur. Priv.*, pp. 305–316, 2010, doi: 10.1109/SP.2010.25.

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

[7] S. Revathi and A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection," *Int. J. Eng. Res. Technol.*, 2013.

[8] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for Network Intrusion Detection in Software Defined Networking," *Proc. - 2016 Int. Conf. Wirel. Networks Mob. Commun. WINCOM 2016 Green Commun. Netw.*, pp. 258–263, Dec. 2016, doi: 10.1109/WINCOM.2016.7777224.

[9] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," *2018 4th IEEE Conf. Netw. Softwarization Work. NetSoft 2018*, pp. 462–469, Sep. 2018, doi: 10.1109/NETSOFT.2018.8460090.

[10] [I. I. Kurochkin and S. S. Volkov, "Using GRU based deep neural network for intrusion detection in software-defined networks," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 927, no. 1, Sep. 2020, doi: 10.1088/1757-899X/927/1/012035.

[11] "NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB." https://www.unb.ca/cic/datasets/nsl.html (accessed Apr. 25, 2022).

[12] "KDD Cup 1999 Data." http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed Apr. 25, 2022).

[13] A. Prabhakaran, V. Kumar Chaurasiya, S. Singh, and S. Yadav, "An optimized deep learning framework for network intrusion detection system (NIDS)," *2020 Int. Conf. Eng. Telecommun. En T 2020*, Nov. 2020, doi: 10.1109/ENT50437.2020.9431266.

[14] A. H. Janabi, T. Kanakis, and M. Johnson, "Convolutional Neural Network Based Algorithm for Early Warning Proactive System Security in Software Defined Networks," *IEEE Access*, vol. 10, pp. 14301–14310, 2022, doi: 10.1109/ACCESS.2022.3148134.

AUTHORS

Mhmood Radhi Hadi is a master's student of Computer engineering at Karabük University, Turkey. Before joining Karabük university in 2020, he was getting BSc degree in 2019 from Networking engineering at Iraqi University, Iraq. His research interests are Network security using AI/ML/DL, Software-defined Network, wireless and communication, Intelligent system, Blockchain.

Adnan Saher Mohammed received his B.Sc. degree in computer engineering technology in 1999 from Northern technical university, Mosul, Iraq. In 2012 obtained an M.Sc. degree in communication and computer network engineering from UNITEN University, Kuala Lampur, Malaysia, and in 2017 received Ph.D. degree from Ankara Yildirim Beyazit University, Ankara, Turkey. He is currently an assistant professor at Karabük University, Turkey. His research interests include computer networks, Algorithms, and Artificial Intelligence.