Deep Learning-based DDoS Detection in Network Traffic Data

Original Scientific Paper

Teeb Hussein Hadi
Middle Technical University,
IT Department, Technical College of Management, Baghdad, Iraq
eng.teebhussien@mtu.edu.iq

Abstract *– In today's society, the cloud is essential for communication since it allows access to important information anytime and anywhere. However, cloud services also attract hackers who want to exploit online details. This has caused significant changes in the cyber-attack landscape. Distributed Denial of Service (DDoS) is the most common attack. Traditional tools like firewalls and encryption can mitigate these risks, but new models are needed to cope with the changing nature of cyber-attacks. Detecting DDoS attacks is particularly challenging since network traffic data is complex and often contains unnecessary features. To address this, a new approach is proposed using Denoising AutoEncoder (DAE) and a Convolutional Neural Network (CNN) for feature selection and classification. The NSL-KDD dataset is used to evaluate the performance of this new model with three main steps: Data Preprocessing, Hyper-parameter Optimization, and Classification. Our method performed better in all four metrics, such as Accuracy, Recall, Precision, and F1-score, with rates of 97.7, 98.1, 97.7, and 97.8, respectively. The multiclass classification detection rate for DOS was 100%. Similarly, the detection rates for Probe, R2L, and U2R were 98%, 95%, and 80%, respectively. Python version 3.6 with Keras 2.2.4 and TensorFlow Engine was used in this paper.*

Keywords: *Network security, DOS, DAE, CNN, Multiclass classification, Deep Learning*

Received: December 15, 2023; Received in revised form: March 4, 2024; Accepted: March 5, 2024

1. INTRODUCTION

Today's interconnected society has revolutionized communication through the advent of IoT services, making vast amounts of information readily accessible online, anytime and from anywhere. Regrettably, this accessibility also exposes the data to cyber-attacks, capitalizing on vulnerabilities that are either unknown or capable of circumventing existing security measures. An effective solution for safeguarding network integrity is the deployment of Intrusion Detection Systems (IDS) [1-3].

IDSs can be categorized in various ways, with one common classification based on their detection method. This categorization divides IDSs into two primary types: signature-based or misuse detection and anomaly-based detection. Signature-based IDSs compare data points with known signatures and trigger an alarm upon detection of a match [4,5]. Conversely, an anomaly-based IDS establishes a pattern from normal traffic and flags any deviation from this pattern as an abnormal transaction. Both methods possess distinct advantages and drawbacks. While signature-based IDSs excel at identifying known attacks, they necessitate frequent manual updates to their signature database. On the other hand, anomaly-based IDSs are adept at uncovering unknown attacks but often produce a plethora of false alarms. Contemporary techniques such as Deep Learning (DL) and Deep Neural Networks (DNN) are increasingly utilized to mitigate these limitations. DL can autonomously learn features and minimize false alarms [6-9].

In this study, we introduce a Convolutional Neural Network (CNN) architecture into our intrusion detection system to identify attacks. Our objective is to classify all four attack categories and subsequently prioritize the detection of Denial of Service (DOS) attacks. Before further processing to reduce data dimensions, we employed a Denoising AutoEncoder (DAE) to select an optimal feature set. We evaluated our model using the NSL-KDD dataset, a refined version of the KDDCup99 and one of the most commonly employed datasets in this domain. Developed by the Defense Advanced Research Projects Agency (DARPA), the KDDCup99 dataset is a benchmark for intrusion detection studies.

While prior research has primarily focused on distinguishing between different attack types, our study proposes a novel approach to binary and multiclass

classification. This approach integrates DAE and CNN for feature selection and classification, respectively. To demonstrate the efficacy of our methodology, we compared the multiclass classification results with those of three previous studies. Our method outperformed existing approaches across all four metrics, including Accuracy, Recall, Precision, and F1-score, as evaluated on the NSL-KDD dataset.

2. NETWORK SECURITY

Network security refers to the different mechanisms and techniques to prevent unauthorized access to digital assets in a network environment. Its main objective is to establish a set of practices that comply with the CIA triad, which stands for confidentiality, integrity, and availability and is the foundation of any security program in an organization [10-12].

This paper is organized as follows: Section 3 describes the dataset, Section 4 presents related work, Section 5 explains the research methodology, Section 6 covers the experimental results and analysis, and Section 7 concludes with future work recommendations.

3. DATASET DESCRIPTIONS

3.1. NSL-KDD

The KDDCup99 is older and has unnecessary data points, which leads to model performance in accuracy while detecting intrusions in an IDS. This issue has been resolved in the refined version of KDDCup99, NSL-KDD. The NSL-KDD is one of the most commonly used datasets in the domain of IDSs. In this work, KDDTrain+. TXT and KDDTest+. TXT files, which have 125,973 and 22,544 records, respectively, are considered. The total number of features in NSL-KDD is 41, with the data types nominal, binary, and numeric. It has four major categories of attacks, which are R2L, U2R, Probe, and DoS, in addition to the Normal class [13-16].

Table 1. Provides a list of features for the dataset

Feature and type	Feature and type	Feature and type
[Duration]=num	[Su Attempted]=bin	[Same Sry Rate]=num
[Protocol Type]=nom	[Num Root]=num	[Diff Sry Rate]=num
[Service]=nom	[Num File Creations]=num	[Sry Diff Host Rate]=nunn
[Flag]=nom	[Num Shells]=num	[Dst Host Count]=num
[Src Bytes]=num	[Num Access Files]=num	[Dst Host Sry Count]=num
[Dst Bytes]=num	[Num Outbound Cmds]=nunn	[Dst Host Same Sry Rate]=num
[Land]=bin	[Is Hot Logins]=bin	[Dst Host Diff Sry Rate]=num
[Wrong Fragment]=num	[Is Guest Login]=bin	[Dst Host Same Srv Rate]=num
[Urgent]=num	[Count]=num	[Dst Host Srv Diff Host Rate]=num
[Hot]=num	[Srv Count]=num	[Dst Host Serror Rate]=num
[Num Failed Logins]=num	[Serror Rate]=num	[Dst Host Srv Diff Host Rate]=num
[Logged In]=bin	[Srv Serror Rate]=num	[Dst Host Serror Rate]=num
[Num Compromised]=num	[Rerror Rate]=num	[Dst Host Srv Rerror Rate]=num
[Root Shell]=bin	[SR/ Rerror Rate]=num	[Label]=nom

4. RELATED WORKS

In [17], the author utilized the NSL-KDD dataset to assess the efficacy of various classification algorithms in detecting abnormalities in network traffic patterns. Their study has yielded valuable insights into the relationship between protocols and network attacks. Their model improves the accuracy of intrusion detection systems and introduces a new research direction in this field. In [18,19], the authors investigated Deep Learning (DL) algorithms to be highly effective in solving various problems across different domains, such as Long Short-Term Memory (LSTM) and Fully Connected Neural Networks (FCNN) that used to categorize benign and malicious connections in intrusion datasets. To achieve a more accurate classification of multi-class assault patterns, They proposed a deep learning model that produces more precise classifications when applied to five-class issues. The model achieves an accuracy of 99.99% when tested on the KDDCup99 dataset and 99.95% on the NSL-KDD dataset. Our model secures the maximum output on both datasets.

In [20], the authors combined two feature selection approaches using LDA and CCA with seven different classifiers: Naive Bayes, Random Tree, Rep-tree, Random Forest, Random Committee, Bagging Randomizable, and Filtered. They concluded that LDA feature selection with Random Tree performed best among the various combinations of feature selection and classifiers. Utilizing LDA and the Random Tree algorithm in anomaly detection was found to be faster and more effective than other methods. Moreover, the accuracy of the Random Tree algorithm surpasses that of different algorithms. This method accurately distinguishes between normal data and various types of attacks. The accuracy of the approach can be further enhanced by employing feature reduction techniques. Based on these findings, it can be inferred that this approach excels in speed, efficiency, and accuracy, especially when implemented on Apache Spark.

In [21], the authors proposed a scenario for backdoor attacks, focusing on the "AlertNet" intrusion detection model and utilizing the NSL-KDD dataset, widely used

in NIDS research. Their study used KL-divergence and OneClassSVM for distribution comparisons to demonstrate resilience against manual inspection by a human expert for outliers. Their experimental results indicated that utilizing decision trees significantly improves the attack's success rate and validated the anomaly regions through KL-divergence, OneClassSVM, and manual inspection.

Authors in [22] proposed a new method to enhance the performance of Intrusion Detection Systems (IDS) on the NSL-KDD dataset. They employed meta-heuristic algorithms and machine-learning techniques for this purpose. Multiple meta-heuristic algorithms were utilized to optimize the hyperparameters of machine learning models, including Random Forest (RF), Support Vector Machine (SVM), Classification and Regression Trees (CART), and Multilayer Perceptron (MLP). The performance of the IDS was evaluated using metrics such as precision, recall, F1-score, and accuracy. Their experimental results demonstrated that the proposed approach outperforms existing techniques in accurately and robustly detecting intrusions.

In [23], the author implemented an IDS framework using Machine Learning (ML) techniques that incorporated various types of Recurrent Neural Networks (RNNs), such as Gated Recurrent Unit (GRU), Long-Short Term Memory (LSTM), and Simple RNN. His results demonstrated that for binary classification tasks using NSL-KDD, XGBoost-LSTM achieved the best performance, with a test accuracy (TAC) of 88.13%, a validation accuracy (VAC) of 99.49%, and a training time of 225.46 seconds. On the other hand, for UNSW-NB15, XGBoost-Simple-RNN was the most efficient model, with a TAC of 87.07%.

In [24], the authors introduced a new approach to enhance the accuracy and efficiency of intrusion detection systems. Their approach utilized Long Short-Term Memory (LSTM) optimized with the Penguin Optimization Algorithm (EPO). Initially, the features underwent preprocessing, including normalization, cleaning, and formatting into numerical format. Subsequently, the Linear Discriminant Analysis (LDA) method was employed to reduce the dimensions of the processed features. Following this, the EPO algorithm was utilized to optimize the size of the hidden units in the LSTM network. Finally, the optimized network was evaluated using the NSL-KDD dataset, a widely recognized benchmark dataset in intrusion detection. Their training and test datasets results were 99.4% and 98.8%, respectively.

Authors in [25,26] decreased the number of features in data using PCA and AutoEncoder. Then, they used Lenet5 CNN for intrusion detection on the KDDCup99 dataset, concluding that the CNN performed better for detecting intrusion on the KDDCup99. According to experimental results, the CNN-IDS model outperforms traditional algorithms in AC, FAR, and timeliness.

5. METHODOLOGY

The general steps of our proposed model are shown in Fig.1. Broadly, it includes data pre-processing, Hyperparameter Optimization, and Classification.

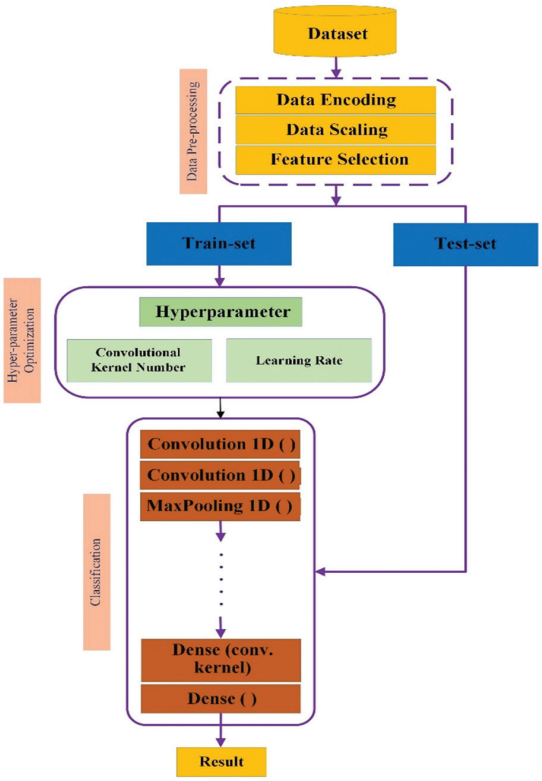


Fig. 1. General steps of the proposed model

5.1. DATA PRE-PROCESSING

The NLS-KDD has some nominal features with many values in each. Those features have to be encoded before any other operation. In this study, a one-hot encoding technique is applied, and then the scaling is performed using the min-max technique, which transforms each feature between 0 and 1. The formula for min-max is given in the equation 1 [27-29].

$$X_a' = \frac{X_a - Min(X)}{Max(X) - Min(X)} \quad (1)$$

Where X_a denotes the original value, X_a' represents the scaled value, $Min(X)$ stands for the minimum value of the feature, and $Max(X)$ gives the maximum value of the feature. The encoding generates many new features in the data, totaling 121 features. A feature selection technique is applied using DAE in the next data preprocessing phase to reduce the number of features. Out of 121 features, only 15 are selected.

5.2. HYPER-PARAMETER OPTIMIZATION

Traditionally, hyperparameters were rarely optimized due to their computational cost requirements. With the advancement of technology, this task is now carried out using modern technologies and powerful algorithms to enhance model performance. This study fo-

cuses on two parameters: convolutional kernel number and learning rate. We provide a range of learning rates, namely 0.03, 0.01, 0.008, 0.006, and 0.004. The convolutional kernel number ranges for optimization are 16-16-32-32, 16-16-64-64, and 32-32-64-64.

5.3. CLASSIFICATION

This study employed a one-dimensional convolutional neural network as a classification model. The classification results in a CNN-based model are directly influenced by the number of convolution kernels and the learning rate [30-33]. We conducted experiments on multiple convolution kernels with different learning rates to obtain the optimal set of parameters. This experiment was carried out on NSL-KDD for multiclass classification. Some significant configurations in the CNN model include loss function = categorical cross-entropy, optimizer = Nadam, pooling = Max Pooling, output activation = softmax, activation function for other layers = ReLU, and dropout parameter = 0.5. We tested three different convolution kernels with five learning rates, as mentioned in section 4.2. The classification metrics used in this paper are Accuracy, Precision, Recall, and F1-score. The calculations for each metric are given by equations 2, 3, 4, and 5, respectively [28-33].

$$Accuracy = \frac{(TN + TP)}{(TP + TN + FP + FN)} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$DR(Recall) = \frac{TP}{(FN + TP)} \quad (4)$$

$$F1 - score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \quad (5)$$

TP stands for True Positive value, TN represents True Negative, FP gives False Positive, and FN denotes False Negative.

6. EXPERIMENTAL RESULTS AND DISCUSSION

The experiment's workstation configuration and tools included a Windows 11 Pro 64-bit operating system with 32 GB RAM and an Intel CPU. The version of Python used was 3.6 with Keras 2.2.4 and Tensorflow Engine. The data was divided into a train set, test set, and validation set with a ratio of 70%, 20%, and 10%, respectively. Table 2 illustrates the appropriate train-test split for the dataset.

Table 2. The number of instances in each class of the NSL-KDD dataset

Class	Train-set (70%)	Test-set (20%)	Validation-set (10%)	Total
Normal	54,153	15,168	7,733	77,054
DoS	37,520	10,508	5,357	53,385
Probe	9,896	2,772	1,409	14,077
R2L	2,637	738	374	3,749
U2R	180	50	22	252
Total	104,386	29,236	14,895	148,517

Intensive comparative analysis has been conducted with the 1D CNN model through various learning rates and convolutional kernel numbers. Table 3 provides the close results.

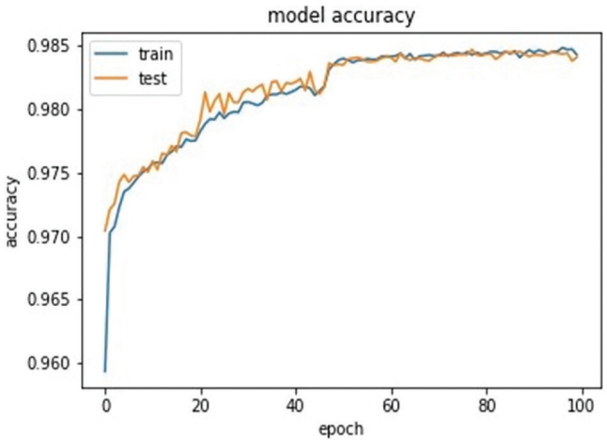
Table 3. Comparison of the proposed model with various numbers of convolution kernels at different learning rates in multi-class classification on the NSL-KDD dataset

Conv. Kernel #	LR	Accuracy %	Precision %	Recall %	F1-score %	Train time in sec.	Test-time in sec.
16-16-32-32	0.03	95.95	97.33	95.95	96.38	76.25	0.82
	0.01	96.49	97.46	96.49	96.80	85.58	0.94
	0.008	96.61	97.49	96.61	96.88	106.17	1.09
	0.006	96.12	97.31	96.12	96.49	104.23	1.18
	0.004	96.21	97.38	96.21	96.58	110.27	1.34
16-16-64-64	0.03	97.52	97.99	97.52	97.67	74.54	1.90
	0.01	97.14	97.77	97.14	97.33	67.24	2.03
	0.008	97.20	97.80	97.20	97.38	65.37	2.17
	0.006	97.48	97.95	97.48	97.62	98.78	2.25
	0.004	97.12	97.79	97.12	97.33	74.49	2.44
32-32-64-64	0.03	97.53	98.04	97.53	97.69	60.85	2.98
	0.01	97.34	97.91	97.34	97.51	67.26	3.20
	0.008	97.68	98.10	97.68	97.81	71.76	3.32
	0.006	97.48	97.96	97.48	97.63	74.33	3.44
	0.004	97.27	97.90	97.27	97.46	101.58	3.59

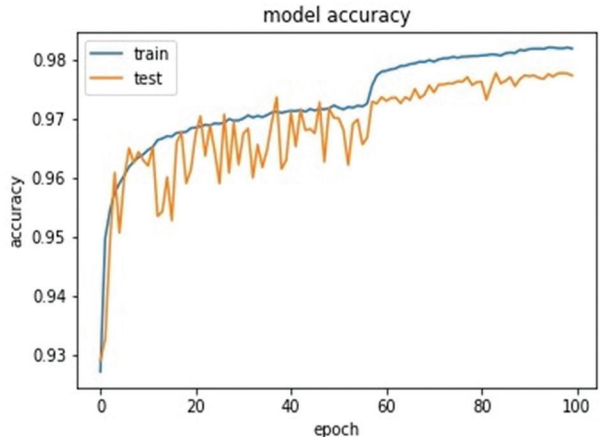
We have observed that the convolution kernel 32-32-64-64, with a learning rate of 0.008, outperforms other configurations in Accuracy, Precision, Recall, and F1-score. However, the training and testing time is minimized with a learning rate of 0.03 with 32-32-64-64 and 16-16-32-32.

Based on the comparison, we can conclude that the convolution kernel 32-32-64-64 with a learning rate of 0.008 is the best among the other configurations. This configuration has been selected as the proposed method for this work. As mentioned in Section 1, this study focuses on binary and multiclass classification.

Fig. 2: (a) depicts the model's accuracy for binary classification across 100 epochs. Similarly, Fig. 2 (b) illustrates the model's accuracy for multiclass classification over the specified epochs. These two figures show that the model's performance improves significantly around 50 epochs and then gradually stabilizes near 100 epochs.



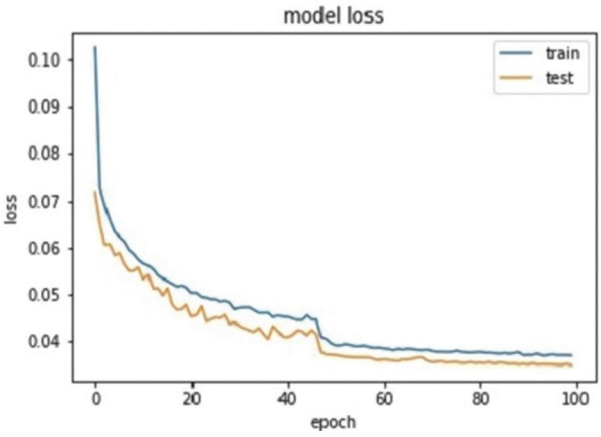
(a)



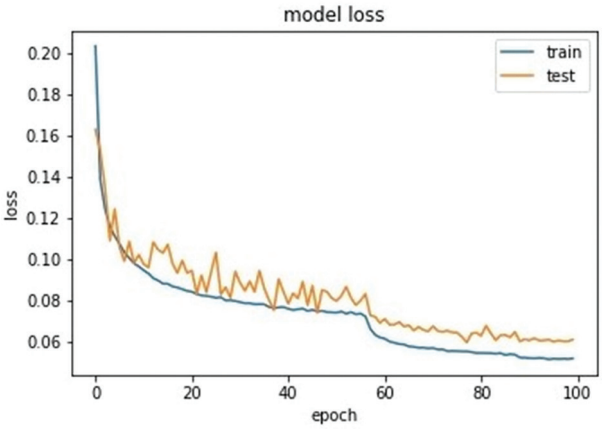
(b)

Fig. 2. Model classification accuracy: (a) Binary (b) Multiclass

Fig. 3: (a) provides a loss of the model for binary classification in the range of 100 epochs. Similarly, Fig. 3: (b) gives the model's loss for multiclass classification in the given epochs. From these two figures, we observe that the loss of the model has dropped around 50 epochs and then slowly stabilized near 100 epochs.



(a)

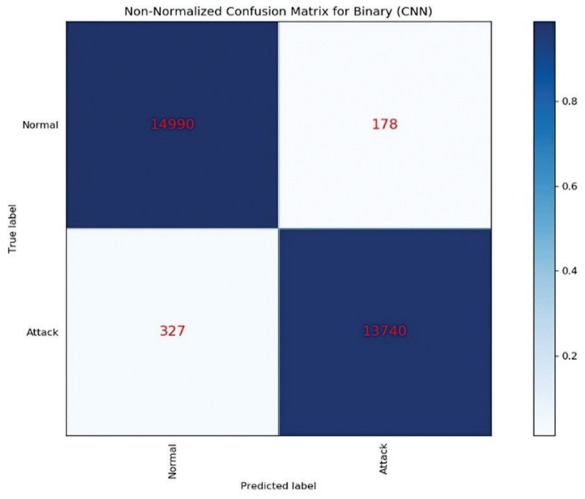


(b)

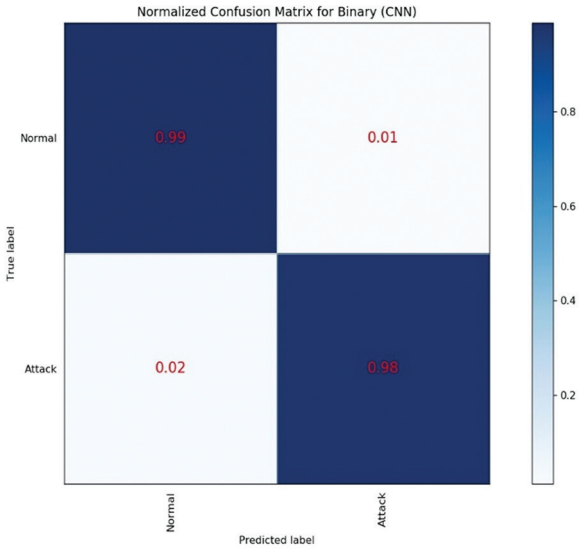
Fig. 3. Model classification loss: (a) Binary (b) Multiclass

Fig. 4: (a) provides a non-normalized confusion matrix of the model for binary classification. Similarly,

Fig. 4: (b) gives the normalized confusion matrix of the model for the same. From these two Figures, we observe that the accuracy performance for binary classification is 0.99 and 0.98 for the normal and attack classes, respectively.

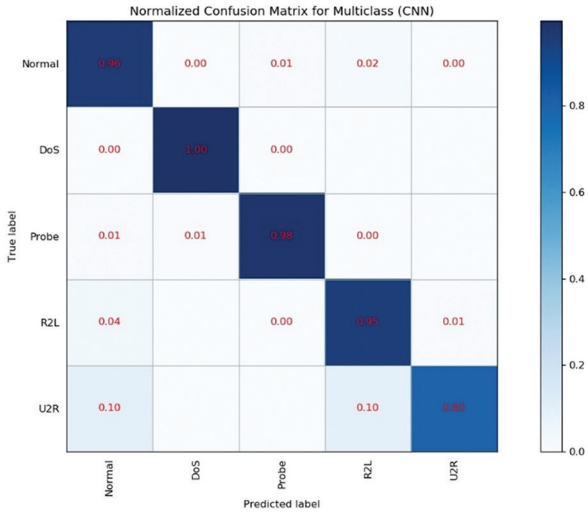


(a)

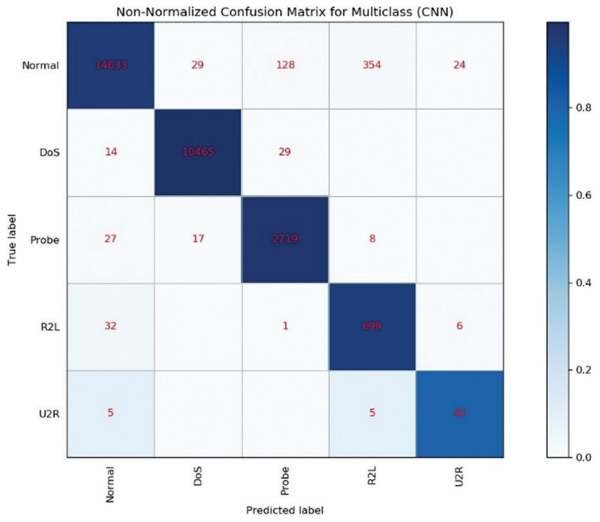


(b)

Fig. 4. Binary classification confusion matrices: (a) Non-normalized (b) Normalized



(a)



(b)

Fig. 5. confusion matrix: (a) Non-normalized (b) Normalized

Fig. 5: (a) presents the non-normalized confusion matrix of the model for multiclass classification. Similarly, Fig. 5 (b) illustrates the normalized confusion matrix of the model for the same task. From these two figures, it is evident that the detection rate performance for multiclass classification is satisfactory. Specifically, in Fig. 5(b), the detection rate for DOS is 100%, while for Probe, R2L, and U2R, the detection rates are 98%, 95%, and 80%, respectively. These results indicate that our approach outperforms three previous works in the domain.

To demonstrate the effectiveness of our method, we have compared our multiclass classification results with some of the previous works in Table 4.

Table 4. Comparison of our results with some of the state-of-the-art

Model	Acc. %	Precision %	DR %	F1-score %
Gaussian–Bernoulli RBM [25]	73.2	62.3	95.1	75.3
ICVAE-DNN [26]	86.0	97.4	77.4	86.3
ID-CVAE [27]	80.1	81.6	80.1	79.1
In this work	**97.7**	**98.1**	**97.7**	**97.8**

7. CONCLUSION AND FUTURE WORK

This research introduces a one-dimensional CNN-based model for intrusion detection. The proposed method comprises three main steps: Data Pre-processing, Hyper-parameter Optimization, and Classification. The number of convolutional kernels and learning rate are two crucial hyperparameters in CNN, so we conducted intensive tuning to identify the best-performing set of parameters. Our experiments showed that the configuration of 32-32-64-64 with a learning rate of 0.008 yielded the best results among all compared configurations. We tested the binary and multiclass classification model on the NSLKDD dataset using Python version 3.6, Keras 2.2.4, and the Tensorflow Engine. The multiclass classification detection rate for DOS was 100%. Similarly, the detection rates for Probe, R2L, and U2R were 98%, 95%, and 80%, respectively. To demonstrate the effectiveness of our method, we compared the multiclass classification results with three previous works. Our method outperformed all four metrics, such as Accuracy, Recall, Precision, and F1-score, with 97.7, 98.1, 97.7, and 97.8 rates, respectively. We plan to conduct further hyperparameter tuning and evaluate the model's performance on different datasets.

8. REFERENCES

[1] S. Mukkamala, G. Janoski, A. Sung, "Intrusion detection using neural networks and support vector machines", Proceedings of the IEEE International Conference on Service-Oriented System Engineering, Oxford, UK, 23-26 August 2021.

[2] M. K. Hooshmand, I. Gad, "Feature selection approach using ensemble learning for network anomaly detection", CAAI Transactions on Intelligent Technology, Vol. 5, No. 4, 2020, pp. 283-293.

[3] J. Kim, J. Kim, H. Kim, M. Shim, E. Choi, "CNN-Based Network Intrusion Detection against Denial-of-Service Attacks", Electronics, Vol. 9, No. 6, 2020, p. 916.

[4] F. Abdulaziz, A. Dahou, M. A. A. Al-cases, S. Lu, M. A. Elaziz, "Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System", Sensors, Vol. 22, No. 1, 2021, p. 140.

[5] L. Dhanabal, S. P. Shantharajah, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, No. 6, 2015, p. 6395.

[6] S. Mohammed, "A Machine Learning-Based Intrusion Detection of DDoS Attack on IoT Devices", International Journal of Advanced Trends in Computer Science and Engineering, Vol. 10, No. 4, 2021, pp. 12-45.

[7] M. K. Hooshmand, M. D. Huchaiah, "Network Intrusion Detection with 1D Convolutional Neural Networks", Digital Technologies Research and Applications, Vol. 1, No. 2, 2022, pp. 25-34.

[8] H. Gharaee, H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM", Proceedings of the 8th International Symposium on Telecommunications, Tehran, Iran, 27-28 September 2016, pp. 139-144.

[9] Sumeet Dua, Xian Du, "Data Mining and Machine Learning in Cybersecurity", 1st Edition, Auerbach Publications, 2016, p.256.

[10] S. Maitra, R. K. Ojha, K. Ghosh, Impact of Convolutional Neural Network Input Parameters on Classification Performance", Proceedings of the 4th International Conference for Convergence in Technology, Mangalore, India, 27-28 October 2018.

[11] H. Zhang, C. Q. Wu, S. Gao, Z. Wang, Y. Xu, Y. Liu, "An Effective Deep Learning Based Scheme for Network Intrusion Detection", Proceedings of the 24th International Conference on Pattern Recognition, Beijing, China, 20-24 August 2018, pp. 682-687.

[12] Z. Ahmad, A. S. Khan, K. Nisar, I. Haider, R. Hassan, M. R. Haque, S. Tarmizi, J. J. P. C. Rodrigues, "Anomaly Detection Using Deep Neural Network for IoT Architecture", Applied Sciences, Vol. 11, No. 15, 2021, pp. 7050-7055.

[13] D. Kumar, R. K. Pateriya, R. K. Gupta, V. Dehalwar, A. Sharma, "DDoS Detection using Deep Learning", Procedia Computer Science, Vol. 218, 2023, pp. 2420-2429.

[14] L. Wen, L. Gao, X. Li, B. Zeng, "Convolutional Neural Network With Automatic Learning Rate Scheduler for Fault Classification", IEEE Transactions on Instrumentation and Measurement, Vol. 70, 2021, pp. 1-12.

[15] A. Chakrabarti, S. Shrivastava, "Intrusion Detection System using Long Short-Term Memory and Fully Connected Neural Network on Kddcup99 and NSL-KDD Dataset", International Journal of Intelligent Systems and Applications in Engineering, Vol.11, No. 9s, 2023, pp. 621-635.

[16] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, K. J. Kim, "A survey of deep learning-based network anomaly detection", Cluster Computing, Vol. 22, No. 1, 2019, pp. 949-961.

[17] R. Sonali, S. Amit, M. Manish, "Intrusion Detection System on KDDCup99 Dataset: A Survey", International Journal of Computer Science and Information Technologies, Vol. 6, No. 4, 2015, pp. 3345-3348.

[18] Y. A. Al-Khassawneh, "An investigation of the Intrusion detection system for the NSL-KDD dataset using machine-learning algorithms", Proceedings of the IEEE International Conference on Electro Information Technology, Romeoville, IL, USA, 18-20 May 2023, pp. 82-87.

[19] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks", IEEE Access, Vol. 7, 2019, pp. 42210-42219.

[20] S. W. A. Alsudani, A. Ghazikhani, "Enhancing Intrusion Detection with LSTM Recurrent Neural Network Optimized by Emperor Penguin Algorithm",

Wasit Journal of Computer and Mathematical Science, Vol. 2, No. 3, 2023, pp. 69-80.

[21] J. Jang, Y. An, D. Kim, D. Cho, "Feature Importance-Based Backdoor Attack in NSL-KDD", Electronics, Vol. 12, No. 24, 2023, p. 4953.

[22] H. G. Ahmed, I. E. Samir, E. K. Ayman, "A Proposed Model for Predicting Employee Turnover of Information Technology Specialists Using Data Mining Techniques", International Journal of Electrical and Computer Engineering Systems, Vol. 12, No. 2, 2021, pp. 113-121.

[23] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework", Computer Communications, Vol. 199, 2023, pp. 113-125.

[24] K. Dinesh, D. Kalaivani, "Enhancing Performance of Intrusion Detection System in the NSL-KDD Dataset using Meta-Heuristic and Machine Learning Algorithms-Design thinking approach", Proceedings of the International Conference on Sustainable Computing and Smart Systems, Coimbatore, India, 14-16 July 2023, pp. 139-144.

[25] Sowmya, T. M. Anita, "An Intelligent Hybrid GA-PI Feature Selection Technique for Network Intrusion Detection Systems", International Journal of Intelligent Systems and Applications in Engineering, Vol. 11, No. 7s, 2023, pp. 718-731.

[26] L. Y. Ahmed, M. M. Hamdy, H. Mahmoud, "Improved DDoS Detection Utilizing Deep Neural Networks and Feedforward Neural Networks as Auto Encoder", Future Internet, Vol. 14, No. 8, 2022, pp. 240-248.

[27] I. Rawaa, T. Abeer, F. Nidaa, "Hybrid Feature Selection Approach to Improve the Deep Neural Network on New Flow-Based Dataset for NIDS," Wasit Journal of Computer and Mathematics Science, Vol. 1, No. 1, 2021, pp. 49-61.

[28] J. Man, G. Sun, "A residual learning-based network intrusion detection system", Security and Communication Networks, Vol. 18, No. 1, 2021, pp. 56-89.

[29] P. Dahiya, D. K. Srivastava, "Network Intrusion Detection in Big Dataset Using Spark", Procedia Computer Science, Vol. 132, 2018, pp. 253-262.

[30] H. Esra'a, A. Hadeel, S. Rizik, A. Orieb, "Hybrid Feature Selection Method for Intrusion Detection Systems Based on an Improved Intelligent Water Drop Algorithm", Cybernetics and Information Technologies, Vol. 22, No. 4, 2022, pp. 73-90.

[31] M. Sheikhan, Z. Jadidi, A. Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping", Neural Computing and Applications, Vol. 21, No. 6, 2012, pp. 1185-1190.

[32] M. Idhammad, K. Afdel, M. Belouch, "DoS Detection Method based on Artificial Neural Networks", International Journal of Advanced Computer Science and Applications, Vol. 8, No. 4, 2017, pp. 465-471.

[33] Y. Fu, Y. Du, Z. Cao, Q. Li, W. Xiang, "A Deep Learning Model for Network Intrusion Detection with Imbalanced Data", Electronics, Vol. 11, No. 6, 2022, pp. 898-904.