The IEEE Region 10 Symposium (TENSYMP)
Netaji Subhas University of Technology, New Delhi, September 27-29, 2024

A Deep Learning Approach for Network Intrusion Classification

Mahbubul Haq Bhuiyan[1], Khorshed Alam[1], Kamrul Islam Shahin[2] and Dewan Md. Farid[1]
[1]Department of Computer Science and Engineering, United International University,
United City, Madani Avenue, Badda, Dhaka 1212, Bangladesh
[2]Software Engineering, The Maersk Mc-Kinney Moller Institute,
University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark
Email: {mahbubul.haq39, mohdkhurshed120}@gmail.com, kish@mmmi.sdu.dk, dewanfarid@cse.uiu.ac.bd

***Abstract*—A Network Intrusion Detection System (NIDS) serves as a sentinel for safeguarding data integrity. It watches over computer networks, looking out for and stopping threats that can sneak past normal defenses like malware and hackers. Deep Learning (DL) techniques offer a promising avenue for analyzing raw network data to uncover subtle patterns indicative of intrusion attempts. In this study, we address a critical research gap by developing a Deep Neural Network (DNN) model tailored for efficient detection of stealthy and polymorphic variants while mitigating false positives. Leveraging the NF-ToN-loT dataset, the proposed model achieves impressive performance metrics on test data, with an accuracy of 0.99, precision of 0.98, recall of 0.99, and F1-score of 0.99. To comprehensively assess the robustness of the proposed model, we use a multi-dataset validation strategy. The model is retrained and evaluated on established benchmark datasets, including NF-BoT-loT, NF-UNSW-NB15, and NF-UNSW-NB15-v2, demonstrating exceptional performance. Furthermore, to ensure the significance of our contribution, we compare our model against previously well-established architectures such as CNN+BiLSTM, DNN, GRU+RNN, and CNN+LSTM. Utilizing the NF-ToN-loT dataset as a common ground, the proposed model demonstrably outperforms these prior models, highlighting its efficacy and advancement in the field. Additionally, we conduct an ablation study to dissect the components of the DNN model, shedding light on their individual contributions towards detecting malware traffic and offering insights for optimizing future NIDS models in the cybersecurity domain.**

Deep Learning, Intrusion Classification, low footprint attacks, Learning from Data

I. Introduction

A Network Intrusion Detection System (NIDS) functions as a cyber sentry, safeguarding data integrity. It diligently monitors computer networks, alerting to and preventing threats that may bypass typical defenses, such as malware and hackers [1]. Deep Learning (DL) can significantly enhance the development of NIDS by using its capability to analyze vast amounts of complex data with minimal human intervention. DL models, particularly Deep Neural Networks (DNNs) can effectively extract intricate patterns and features from raw network traffic data. By training on large datasets of labeled network traffic, Deep Learning models can learn to distinguish between normal and anomalous behavior, enabling them to identify various types of intrusions, including malware attacks, unauthorized access attempts, and denial-of-service attacks. Recent research on NIDS faces several critical gaps. NIDS are a vital defense mechanism, but there are ongoing challenges in keeping them effective. A key research gap is the reliance on outdated datasets for training and evaluation. These datasets often don't reflect the ever-changing tactics of attackers. Additionally, the focus on achieving high accuracy metrics in research may not translate to real-world performance. NIDS need to be adaptable to new threats and network environments. Furthermore, securing emerging technologies like smart grids poses a challenge, as traditional NIDS may not be suited to their specific needs. In this paper, we solved some limitations which is dicussed in Section II(A). We propose a DNN based DL model for intrusion detection from network data. We summarize the contributions as below.

1) The study introduces an innovative DNN model specifically designed for efficient detection of stealthy and polymorphic variants of network intrusions. This model is aimed at mitigating false positives, a crucial aspect in network security.
2) To comprehensively evaluate the robustness of the model, a multi-dataset validation strategy is used shown in Table IV. The model is retrained and evaluated on established benchmark datasets, including NF-BoT-loT and NF-UNSW-NB15, showcasing exceptional performance across different datasets.
3) The study compares the proposed model with previously established architectures such as CNN+BiLSTM, DHN, GRU+RNN, and CNN+LSTM. Utilizing the NF-ToN-loT dataset as a common ground for comparison, the proposed model outperforms these prior architectures, highlighting its efficacy and advancement in the field shown in Table III.
4) An ablation study is conducted to dissect the components of the DNN model shown in Table II. This analysis provides insights into the individual contributions of each component towards detecting malware traffic, offering valuable information for optimizing future NIDS models in the cybersecurity domain.

II. Previous Work

Various techniques are available for detecting network intrusion activity. In their study [2], the authors explored the application of machine learning algorithms like Naïve Bayes, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) for intrusion detection. The study evaluated the performance of these algorithms on datasets like NSL-KDD

2024 IEEE Region 10 Symposium (TENSYMP) | 979-8-3503-6486-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/TENSYMP61132.2024.10752251

and UNSW-NB15. No prior study comparison is found in the paper. However, the study [3] solved the limitation by providing comprehensive prior study comparison. Their suggested approach involves generating a fresh training dataset for Generative Adversarial Network (GAN) by leveraging misclassified data sourced from a bespoke training dataset via an LSTM-DNN model trained on the original dataset. There is hardware dependency and accuracy needs to be further improved, compared to the conventional approaches.

Convolutional Deep Belief Networks (CDBN) based model proposed by [4] for intrusion detection in wireless networks using . However, CDBN model can be computationally intensive due to their complex architecture and the need for significant computational resources for training and inference. In [5]'s work, the authors introduced a novel hybrid model combining Convolutional Neural Network (CNN) and bidirectional long short-term memory (BiLSTM) networks, aimed at improving network intrusion detection through both binary and multiclass classification tasks. L2 regularizer and dropout (0.5) techniques have been used in the training model. They achieve an average accuracy of 84.42% on NF-UNSW-NB15. The paper [6] used SMOTE oversampling prior to feature selection, utilizes One-Hot encoding for numerical conversion, utilized the Random Forest (RF) algorithm for binary classification on the training dataset, and adopts a CNN with Relu activation function for mapping data to hidden layer feature dimension, culminating in Softmax activation for quaternion classification results. No prior study comparison is found in the paper. Authors from [7] proposed the Hierarchical Clustering Algorithm (HCA) based decision tree for network traffic data. The paper may not extensively compare the proposed HC-DTTSVM method with a wide range of existing network intrusion detection methods, potentially limiting the understanding of its performance in relation to other approaches.

A. Dealing with limitations

Previous studies like [2] and [6] lacked comprehensive comparisons with other models. In contrast, our paper conducts a thorough comparison by evaluating our proposed DNN model against well-established architectures such as CNN+BiLSTM, DHN, GRU+RNN, and CNN+LSTM. This comparison provides a clear understanding of the efficacy and advancement of our model in the field shown in Table III. The study by [4] proposed a deep learning-based mechanism for intrusion detection but noted computational intensity due to the complex architecture of Convolutional Deep Belief Networks (CDBN). Our paper addresses this limitation by developing a DNN model tailored for efficient detection of stealthy and polymorphic variants while mitigating false positives. This suggests that our model achieves impressive performance without the excessive computational resources required by CDBNs as shown in Table I. While some previous studies reported accuracy metrics, our paper provides a comprehensive assessment of model performance by including accuracy, precision, recall, and F1-score metrics. Moreover, we validate the robustness of our model by retraining and evaluating it on multiple benchmark datasets, including NF-BoT-loT, NF-UNSW-NB15, and NF-UNSW-NB15-v2 shown in Table IV. Previous studies like [5] utilized techniques like L2 regularization and dropout but might not have provided insights into the individual contributions of model components towards detecting malware traffic. Our paper conducts an ablation study to dissect the components of the DNN model, offering insights for optimizing future NIDS models in the cybersecurity domain shown in Table II. This provides valuable guidance for researchers and practitioners aiming to improve intrusion detection systems.

III. Proposed NIDS Framework

In this section, we describe the development process of our intrusion Detection model, transparently laying out the rationale of this paper, as illustrated in Figure 1. The proposed methodology is based on three distinct steps.

Database Formation: In this step, we collected four open source datasets from University of Queensland to build our DNN model.

Data Pre-processing: Some pre-processing is done before model development, in order to fed correct data to model to gain maximum out of it.

Model Development: In this step, we demonstrate the development of DNN model, reason behind choosing this algorithm, parameters and so on.

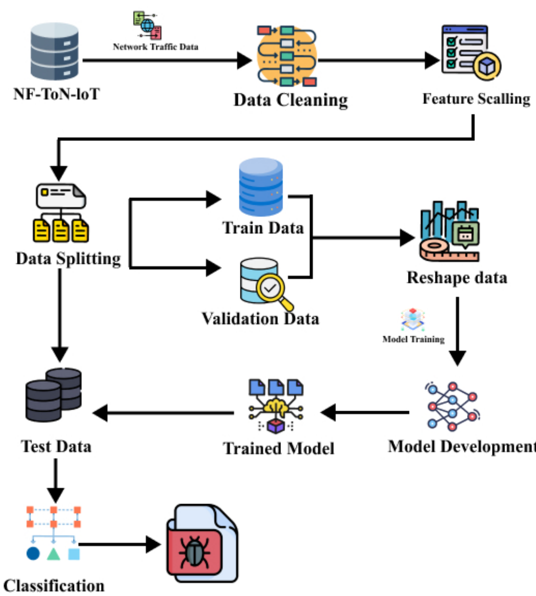


Fig. 1: Framework of Proposed Network Intrusion Detector.

A. Database Formation

We collected datasets from open-source NIDS datasets from the University of Queensland, Australia, including NF-ToN-loT, NF-BoT-loT, NF-UNSW-NB15, and NF-UNSW-NB15-v2. The NF-UNSW-NB15 dataset comprises 1,623,118 data flows, with 4.46% categorized as attacks. NF-ToN-IoT contains 1,379,274 data flows, with 80.4% as attack samples. NF-BoT-IoT, with 600,100 flows, has 97.69% attacks. NF-UNSW-NB15-v2 includes 2,390,275 flows, with 3.98% attacks. These datasets cover a range of attack types, such as DoS, DDoS, Reconnaissance, and Exploits, targeting network security in both traditional and IoT environments. Features were extracted from pcap files to label flows with their respective attack categories. We use NF-ToN-loT dataset as primary dataset and rest of the datasets are for multi-dataset validation strategy.

B. Data Preprocessing

The preprocessing steps described below are crucial for preparing the NF-ToN-IoT dataset for training a DNN model for classification tasks. Initially, the dataset is loaded into a pandas DataFrame, facilitating subsequent data manipulation. Two columns, 'IPV4_SRC_ADDR' and 'IPV4_DST_ADDR', which likely represent IP addresses, are dropped from the dataset as they may not directly contribute to the classification task, focusing the model on relevant network flow features. The 'Label' column, is the target variable, undergoes one-hot encoding to transform categorical labels into a format suitable for neural network training. Additionally, feature scaling is performed using StandardScaler to standardize features, ensuring they have a mean of 0 and a standard deviation of 1. This scaling technique helps in improving convergence speed and performance, particularly when features possess varying scales. The dataset is then reshaped to fit the requirements of the DNN model, with the dimensions adjusted to include the number of channels. Finally, the reshaped data is split into training and testing sets, enabling the evaluation of the CNN model's performance on unseen data. These preprocessing steps collectively enhance the effectiveness of the CNN model by appropriately formatting the data for training while removing irrelevant features and facilitating model evaluation. We use 80% data for training and 20% data for testing purpose.

C. Model Development

We propose a DNN based deep learning architecture for our network intrustion detection framework which achieves noteworthy performance on NF-ToN-loT, NF-BoT-loT, NF-UNSW-NB15, and NF-UNSW-NB15-v2. In Table I and Fig. 2, we demonstrated the DNN architecture of this work. Let us discuss about the layers and rationale of using these layers to build this model.

The DNN architecture for intrusion detection is designed with careful consideration of the number of layers, neurons, optimizer, and learning rate to balance complexity, efficiency, and performance. The model consists of an input layer tailored to the number of features in the training data, followed by four hidden layers with decreasing neuron counts (64, 32, 32, and 16) to progressively condense and distill feature representations. Each hidden layer uses the ReLU activation function (Eq. 2) to effectively mitigate the vanishing gradient problem and accelerate learning. To prevent overfitting, dropout layers with a 20% dropout rate are included after each hidden

TABLE I: Proposed DNN Architecture.

Layer (type)	Output Shape	Param
dense_5 (Dense)	(None, 64)	704
dropout_3 (Dropout)	(None, 64)	0
dense_6 (Dense)	(None, 32)	2080
dropout_4 (Dropout)	(None, 32)	0
dense_7 (Dense)	(None, 32)	1056
dropout_5 (Dropout)	(None, 32)	0
dense_8 (Dense)	(None, 16)	528
dense_9 (Dense)	(None, 2)	34
Total Params		**4402 (17.20 KB)**
Trainable Params		**4402 (17.20 KB)**
Non-trainable Params		**0 (0.00 Byte)**

layer, encouraging the model to learn robust features. The output layer euses a softmax activation function to produce a probability distribution over the classes, suitable for multi-class classification tasks.

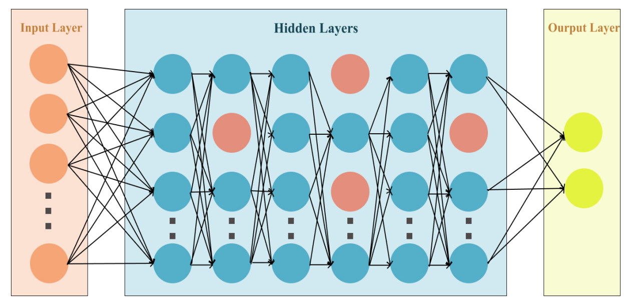


Fig. 2: Custom DNN Architecture.

Fully connected layers, also known as dense layers, are pivotal in processing the flattened output from preceding layers. Their primary function is to conduct high-level feature extraction and classification. Through dense layers, the network integrates the extracted features from earlier layers to make predictions, discerning between malware and benign activities within network traffic. Utilizing multiple dense layers allows the model to grasp intricate relationships between input features and output labels, thus bolstering its discriminatory capabilities. Let x represent the input data, w denote the weight, b stand for the bias, and σ symbolize the activation function that operates element-wise on the summation of weighted inputs and biases shown in Eq. (1).

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{1}$$

Dropout layers are a vital component in regularization, as they randomly deactivate a portion of input units during training. This strategy effectively combats overfitting and enhances the model's capacity for generalization. Overfitting arises when the model memorizes noise rather than discerning genuine patterns within the training data. By incorporating dropout layers, the model is compelled to acquire more resilient and comprehensive representations of the data, thereby

TABLE II: Ablation Study of Proposed Model (30 Epochs).

Optimizer	Dataset	Learning Rate	Accuracy	Precision	Recall	F1 Score
Adam	NF-ToN-loT	0.001	0.9904	0.9894	0.9987	0.9940
Adam	NF-ToN-loT	0.0001	0.9904	0.9895	0.9987	0.9941
Adam	NF-ToN-loT	0.00001	0.9873	0.9850	0.9994	0.9921
RMSprop	NF-ToN-loT	0.001	0.9844	0.9810	0.9999	0.9904
RMSprop	NF-ToN-loT	0.0001	0.9870	0.9845	0.9994	0.9919
RMSprop	NF-ToN-loT	0.00001	0.9869	0.9844	0.9995	0.9919
Adagrad	NF-ToN-loT	0.001	0.9869	0.9844	0.9995	0.9919
Adagrad	NF-ToN-loT	0.0001	0.9707	0.9649	0.9999	0.9821
Adagrad	NF-ToN-loT	0.00001	0.8037	0.8037	0.9015	0.8912
SGD	NF-ToN-loT	0.001	0.9867	0.9854	0.9982	0.9917
SGD	NF-ToN-loT	0.0001	0.9811	0.9773	0.9997	0.9884
SGD	NF-ToN-loT	0.00001	0.9217	0.9568	0.9452	0.9510

refining its capability to identify malware activity in novel network traffic.

$$ReLU(x) = max(0, x) \tag{2}$$

The output layer produces the final predictions of the model, indicating whether the network traffic is classified as Malware or benign. The output layer employs an activation function (in this case, softmax) to transform the model's raw output into a probability score or binary classification decision. This allows the model to make informed decisions about the presence of malwar activity based on the learned representations from earlier layers.

The Adam optimizer, known for its adaptive learning rate and momentum properties, is selected for its ability to improve convergence speed and stability. A learning rate of 0.0001 is chosen to ensure gradual and precise updates to the model's weights, avoiding the risk of overshooting the minima of the loss function. The model is compiled using categorical cross-entropy as the loss function, ideal for multi-class classification, and accuracy as the performance metric. This combination of architectural choices aims to create a robust, generalizable model capable of effectively identifying intrusions with high accuracy and minimal overfitting.

(a) Accuracy of Our Model (lr=0.0001, opt=adam)

(b) Loss of Our Model (lr=0.0001, opt=adam)

(c) Accuracy of Our Model (lr=0.0001, opt=Rmsprop)

(d) Loss of Our Model (lr=0.0001, opt=Rmsprop)

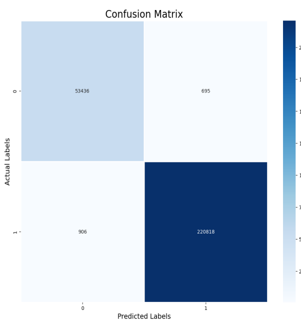
Fig. 3: The Accuracy and Loss Metrics Result for Proposed Model on NF-ToN-IoT Dataset.

IV. Performance Analysis

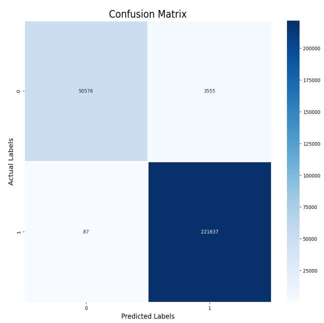
Upon completing model training, it demonstrated a remarkable 99% accuracy on both training and testing datasets. Nevertheless, relying solely on accuracy may not provide a comprehensive evaluation of the model's performance. Given the nature of this classification task, it is imperative to consider additional metrics such as precision, recall, F1-score, and confusion matrix. These performance indicators are detailed in Tables II, offering a more nuanced assessment of the model's effectiveness.

The table II provides results from an ablation study conducted on a DNN-based intrusion detection system, highlighting the impact of various optimizers and learning rates on the NF-ToN-IoT dataset. The optimizers tested include Adam, RMSprop, Adagrad, and SGD (Stochastic Gradient Descent), with learning rates set at 0.001, 0.0001, and 0.00001. Performance metrics evaluated are accuracy, precision, recall, and F1 score. From the table, it's evident that the Adam optimizer yields consistently high results across all metrics, particularly at a learning rate of 0.0001, suggesting an optimal balance between learning speed and model performance. In contrast, the performance drops significantly for Adagrad and SGD when the learning rate is decreased to 0.00001, as seen in the substantial declines in accuracy, precision, recall, and F1 score. This study demonstrates how optimizer choice and learning rate tuning are crucial for optimizing model performance, with Adam appearing as the most effective optimizer for this specific intrusion detection task on the NF-ToN-IoT dataset.

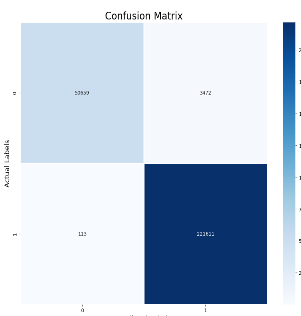
As it is a classification problem, we choose accuracy, precision, recall and f1 score as performance matrix. Precision evaluates the model's ability to accurately identify genuine instances of malware from all instances it predicts as malware. A high precision indicates fewer false positive predictions, which is crucial for minimizing false alarms in trojan detection systems. On the other hand, recall, also known as the true positive rate, measures how well the model can correctly detect actual malware instances in network traffic data. In our context, it showcases the model's effectiveness in identifying a significant portion of malware instances, thereby lowering the chances of undetected malware activity. A high recall implies efficient intrusion detection, thereby reducing the risk of overlooking malware instances. The F1 score provides a balanced assessment of the model's intrusion detection accuracy by taking into account both precision and recall, effectively considering false positives and false negatives. In our scenario, it serves as a comprehensive metric for evaluating both precision and recall simultaneously, which is essential for effective intrusion detection.



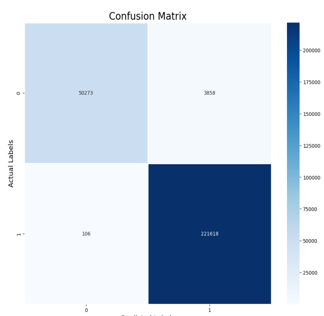
(a) Confusion Matrix (lr=0.0001, opt=adam)



(b) Confusion Matrix (lr=0.001, opt=adam)



(c) Confusion Matrix (lr=0.0001, opt=rmpsprop)



(d) Confusion Matrix (lr=0.001, opt=rmpsprop)

Fig. 4: Confusion Matrix on NF-ToN-IoT Dataset.

A confusion matrix is a performance measurement for machine learning classification problems where output can be two or more classes. The confusion matrix of our proposed model. In figure 4, we illustrated the confusion matrix of our model on NF-ToN-IoT Dataset. A set of experiments are done to ensure contribution. A set of experiments are done to ensure contribution. Our proposed model is evaluated against previously used alternative neural network architectures using NF-ToN-IoT Dataset, and the findings are presented in Table III and Fig. 5.

Table III presents a comparison of the performance metrics for several machine learning models as used in previous studies, with a learning rate (lr) of 0.0001 and the Adam opti-

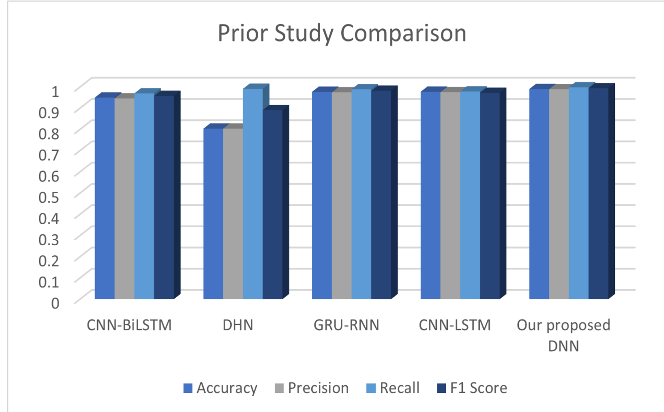


Fig. 5: Performance Comparison with Previously used Models.

TABLE III: Comparison of the performance among most used models from previous studies.

Model	Accuracy	Precision	Recall	F1 Score
DHN [9]	0.8038	0.8038	0.9912	0.8912
CNN-BiLSTM [5], [11], [12]	0.9501	0.9460	0.9698	0.9577
GRU-RNN [13], [14]	0.9772	0.9749	0.9894	0.9821
CNN-LSTM [15], [16]	0.9775	0.9758	0.9788	0.9723
Our work	0.9904	0.9895	0.9987	0.9941

mization algorithm using NF-ToN-IoT Dataset. The table lists different models including Deep Hierarchical Network (DHN), CNN-BiLSTM, GRU-RNN, CNN-LSTM, and the propoed model. It shows that our model performs exceptionally well with an accuracy of 0.9941, precision of 0.9968, recall of 0.9992, and F1 score of 0.9963, suggesting it may be the most effective among those listed.

TABLE IV: Comparison of the performance among other benchmark datasets.

Dataset	Accuracy	Precision	Recall	F1 Score
NF-ToN-loT [17]	0.9904	0.9895	0.9987	0.9941
NF-BoT-loT [18]	0.9902	0.9911	0.9989	0.9950
NF-UNSW-NB15 [19]	0.9861	0.9776	0.9727	0.9847
NF-UNSW-NB15-v2	0.9968	0.9284	0.9786	0.9528

Achieving high accuracy on a single dataset is not sufficient to demonstrate the robustness of a model. Robustness entails the model's ability to maintain its performance across various datasets and under different conditions. To thoroughly evaluate the robustness of our model, we retrained it using three diverse datasets: NF-BoT-IoT, NF-UNSW-NB15, and NF-UNSW-NB15-v2. The results of this comprehensive evaluation are presented in Table IV, showcasing our model's noteworthy performance across these datasets. This approach ensures that our model is not just overfitting to one dataset but is generalizable and reliable in different scenarios.

V. Conclusion

This study presents a significant advancement in network intrusion detection through the development of a Deep Neural Network (DNN) model specifically designed for the efficient and accurate identification of stealthy and polymorphic intrusion attempts. The proposed model achieved an impressive accuracy of 99.04%, with precision at 98.95%, recall at 99.87%, and an F1 score of 99.41%, demonstrating substantial improvements over previously established architectures. Notably, the model outperformed traditional methods in key performance metrics, underscoring its efficacy in reducing false positives and enhancing detection rates.

Looking ahead, the future scope of this research includes exploring the integration of the DNN model with real-time intrusion detection systems, further optimizing the model through advanced techniques such as transfer learning and ensemble methods. Additionally, expanding the dataset to include more diverse and complex attack scenarios will enhance the model's robustness and adaptability to evolving cyber threats. This research sets the groundwork for developing more resilient cybersecurity solutions that can safeguard critical data and infrastructure in increasingly sophisticated digital environments.

References

[1] D. S. Bauer and M. E. Koblentz, "NIDX-an expert system for real-time network intrusion detection," [1988] Proceedings. Computer Networking Symposium, Washington, DC, USA, 1988, pp. 98-106, doi: 10.1109/CNS.1988.4983.

[2] R. Tahri, Y. Balouki, A. Jarrar, and A. Lasbahani, "Intrusion Detection System Using machine learning Algorithms," ITM Web of Conferences, vol. 46. EDP Sciences, p. 02003, 2022. doi: 10.1051/itm-conf/20224602003.

[3] T. Kim and W. Pak, "Early Detection of Network Intrusions Using a GAN-Based One-Class Classifier," in IEEE Access, vol. 10, pp. 119357-119367, 2022, doi: 10.1109/ACCESS.2022.3221400.

[4] L. Yang, J. Li, L. Yin, Z. Sun, Y. Zhao and Z. Li, "Real-Time Intrusion Detection in Wireless Network: A Deep Learning-Based Intelligent Mechanism," in IEEE Access, vol. 8, pp. 170128-170139, 2020, doi: 10.1109/ACCESS.2020.3019973.

[5] R. Ben Said, Z. Sabir and I. Askerzade, "CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection," in IEEE Access, vol. 11, pp. 138732-138747, 2023, doi: 10.1109/ACCESS.2023.3340142.

[10] V. K. Navya, J. Adithi, D. Rudrawal, H. Tailor and N. James, "Intrusion Detection System using Deep Neural Networks (DNN)," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2021, pp. 1-6, doi: 10.1109/ICAECA52838.2021.9675513.

[6] X. Gao, Q. Wu, J. Cai and Q. Li, "A Fusional Intrusion Detection Method Based on the Hierarchical Filtering and Progressive Detection Model," in IEEE Access, vol. 11, pp. 131409-131417, 2023, doi: 10.1109/ACCESS.2023.3335669.

[7] L. Zou, X. Luo, Y. Zhang, X. Yang and X. Wang, "HC-DTTSVM: A Network Intrusion Detection Method Based on Decision Tree Twin Support Vector Machine and Hierarchical Clustering," in IEEE Access, vol. 11, pp. 21404-21416, 2023, doi: 10.1109/ACCESS.2023.3251354.

[8] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a Standard Feature Set for Network Intrusion Detection System Datasets," Mobile Networks and Applications, vol. 27, no. 1. Springer Science and Business Media LLC, pp. 357–370, Nov. 10, 2021. doi: 10.1007/s11036-021-01843-0.

[9] K. Jiang, W. Wang, A. Wang and H. Wu, "Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network," in IEEE Access, vol. 8, pp. 32464-32476, 2020, doi: 10.1109/ACCESS.2020.2973730.

[11] W. Dai, X. Li, W. Ji and S. He, "Network Intrusion Detection Method Based on CNN-BiLSTM-Attention Model," in IEEE Access, vol. 12, pp. 53099-53111, 2024, doi: 10.1109/ACCESS.2024.3384528.

[12] X. Liang, H. Xing and T. Hou, "Network Intrusion Detection Method Based on CGAN and CNN-BiLSTM," 2023 IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI), Harbin, China, 2023, pp. 396-400, doi: 10.1109/ICEMI59194.2023.10270557.

[13] 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 Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, 2018, pp. 202-206, doi: 10.1109/NETSOFT.2018.8460090.

[14] I. I. Kurochkin and S. S. Volkov, "Using GRU based deep neural network for intrusion detection in software-defined networks," IOP Conference Series: Materials Science and Engineering, vol. 927, no. 1. IOP Publishing, p. 012035, Sep. 01, 2020. doi: 10.1088/1757-899x/927/1/012035.

[15] M. Abdallah, N. An Le Khac, H. Jahromi, and A. Delia Jurcut, "A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs," Proceedings of the 16th International Conference on Availability, Reliability and Security. ACM, Aug. 17, 2021. doi: 10.1145/3465481.3469190.

[16] L. Karanam, K. K. Pattanaik and R. Aldmour, "Intrusion Detection Mechanism for Large Scale Networks using CNN-LSTM," 2020 13th International Conference on Developments in eSystems Engineering (DeSE), Liverpool, United Kingdom, 2020, pp. 323-328, doi: 10.1109/DeSE51703.2020.9450732.

[17] M. Sarhan and S. Layeghy, "NF-ToN-IoT." The University of Queensland, 2023. doi: 10.48610/2FA2ED6.

[18] M. Sarhan and S. Layeghy, "NF-BoT-IoT." The University of Queensland, 2023. doi: 10.48610/62E6D80.

[19] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 2015, pp. 1-6, doi: 10.1109/MilCIS.2015.7348942.