A Deep Learning-Based Intrusion Detection System using Refined LSTM for DoS Attack Detection

Mohammad Hiari
Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
m.hyari@ammanu.edu.jo

Yousef Alraba'nah
Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
y.alrabanah@ammanu.edu.jo (corresponding author)

Iyas Qaddara
Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
i.qaddara@ammanu.edu.jo

Received: 15 April 2025 | Revised: 30 May 2025 and 18 June 2025 | Accepted: 21 June 2025

 DOI: https://doi.org/10.48084/etasr.11499

ABSTRACT

The detection of a Denial of Service (DoS) attacks is a key challenge in network security, directly impacting the availability and reliability of networks. Such attacks have to be mitigated by implementing an accurate and timely detection mechanism to ensure the integrity of the network infrastructure. Driven by the shortcomings of conventional attack detection methods and the growing complexity of the network attacks, this work proposes a new customized Long Short-Term Memory (LSTM) model for DoS attacks detection. The proposed deep learning approach utilizes LSTM's strength in learning long-range dependencies in sequential data to model network traffic patterns over time. The effectiveness of the model is evaluated through comparative experiments. The primary outcome is that the proposed LSTM model has improved detection performance across all metrics of precision, recall, F1-score, and accuracy. The results demonstrate that the proposed LSTM architecture is a promising and trustworthy solution for enhancing intrusion detection systems (IDSs) and protecting network systems against DoS attacks.

Keywords-LSTM; DoS; NSL-KDD; intrusion detection; DL

I. INTRODUCTION

Intrusion Detection Systems (IDSs) are a critical component of contemporary cybersecurity strategies, developed to identify and respond to network breaches. The IDS can monitor network traffic, find unusual patterns, and inform the network administrators about probable threats. Due to the explosion of e-commerce, the infrastructure type and amount of traffic have changed. Consequently, the network behavior characteristics are becoming increasingly complex, which poses significant challenges to IDSs. The primary concern pertains to the nature of the traffic, which can serve as an indication of an unknown and therefore unaddressed attack [1]. IDSs are classified into two broad categories: signature-based IDSs and anomaly-based IDSs. The former relies on published patterns of known attacks and, as a result, is capable of recognizing known attacks, but has limited efficiency in detecting new or changing threats. The latter establishes a threshold of normal behavior and notifies any anomaly, which thus allows one to discover those threats that have never been seen before. On the other hand, it should be noted that the growing complexity and volume of network traffic have devastated the efficacy of traditional IDSs. This underscores the necessity for the integration of contemporary technologies [2].

Among the most disruptive threats faced by modern networks are Denial of Service (DoS) attacks, which intend to flood a network or system with excessive traffic, making it unavailable to legitimate users [3]. A more sophisticated variant, the Distributed Denial of Service (DDoS) attack, uses diverse compromised devices to launch a coordinated assault, amplifying its impact and making its effects more challenging. DoS and DDoS attacks can result in the failure of critical infrastructure, resulting in significant financial losses, reputational damage, and operational disruptions [4]. Detecting and mitigating these attacks in real-time is a pressing challenge, particularly as attackers continually refine their methods to evade traditional defenses [5].

Today's models are inadequate in addressing the fast and complex nature of networks compromised by cyberattacks. Consequently, they exhibit low false alarm rate but they also suffer from low detection rates and increased communication and computational costs [6]. The classical methods of malicious detection include encryption, access control mechanisms, firewalls, etc. However, these approaches have inherent limitations that preclude full network protection. Intrusion detection is an essential building block for system administrators in a network, as it enables the monitoring of a variety of malicious [7]. Depending on the technique employed for intrusion detection, IDSs are often considered classifier machines. The IDS is responsible for monitoring all network traffic and categorize it as either normal or malicious. This capability allows IDS to employ machine learning techniques to improve classification accuracy. Various methodologies for developing IDSs have relied on traditional machine learning algorithms, including Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), K-Nearest Neighbors (KNN), and Random Forest (RF). These traditional algorithms have several limitations, especially as intrusions are getting more complex and diverse [8]. Therefore, there is a need for improved learning techniques, particularly those focused on automatically identifying and analyzing intrusion features.

Deep learning has emerged as a disruptive technology in the area of cybersecurity, leveraging multi-layer neural networks to automatically learn complicated patterns and structures from high volumes of data [9]. Deep learning with multiple layers of processing units has demonstrated efficacy in the domains of anomaly detection, malware identification, and attack forecasting. In the context of IDS and DoS detection, deep learning models can process huge volumes of network data, thereby capturing subtle signals of intruders that may not be detected by traditional methods [10]. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTMs) are examples of the popular techniques of deep learning, which have the potential to enhance the efficiency and accuracy of intrusion detection and attack mitigation systems [11]. The integration of deep learning into IDS and DoS detection systems signifies a transformation in the cybersecurity paradigm. Automatic detection systems, in conjunction with a reduction in manual rule-based systems, allow for the deployment of deep learning, which in turn guarantee adaptive and robust defenses [12].

II. LITURATURE REVIEW

Authors in [13] proposed a method for detecting DDoS attacks that uses LSTM networks improved with the utilization of the Singular Value Decomposition (SVD) technique for feature selection. The essential idea of this study is to refine the existing way of detecting DDoS attacks, with the aim of increasing accuracy, reducing false positives, and at the same time handling large-scale network traffic in a more effective way. The method was evaluated using two well-known anomaly detection datasets: UNSW-NB15 and NSL-KDD. Researchers extracted 23 and 20 key attack features from each dataset, respectively, using SVD. The results showed that the LSTM model outperformed traditional machine learning approaches including Naïve Bayes (NB), Decision Tree (DT), and SVMs. The model achieved 94.28% accuracy on the UNSW-NB15 dataset and 90.59% on NSL-KDD, surpassing other models in detection rate, recall, and F1-score.

Authors in [14] suggested a DoS attack detection method based on machine learning that was evaluated using the NSL-KDD dataset by assessing four popular classification algorithms, NB, KNN, DT, and SVM. The authors removed non-DoS attack related records, performed label encoding for categorical features, and standardized the dataset using min-max normalization. The testing results indicated that the DT classifier achieved the best results: 99.89% accuracy, 99.90% recall, 99.91% precision, and a Matthews Correlation Coefficient (MCC) score of 99.96%. On the other hand, NB demonstrated the least efficacy. Therefore, the study concluded that DT classifiers are highly effective for detecting DoS attacks and thus, can be the choice for those systems that require intelligent attack classification.

Authors in [15] proposed a deep learning-based model to detect DDoS attacks in the cloud environment. They focused on addressing the limitations of conventional IDSs, such as delayed convergence and local stagnation. To achieve this objective, the authors employed a hybrid approach, integrating LSTM with Harris Hawks Optimization (HHO) and Particle Swarm Optimization (PSO). The hybrid HHO and PSO method was applied to the selection of features and the tuning of parameters to increase LSTM classification accuracy. They tested the model with the NSL-KDD dataset using performance measurements such as accuracy, precision, recall and F1-score. The proposed hybrid HHO and PSO LSTM model demonstrated an accuracy of 99.53%, which was higher than the accuracy of traditional classifiers, including NB, DT, and SVM. It was shown that deep learning and hybrid optimization improve intrusion detection.

Authors in [16] investigated the effectiveness of various machine learning classifiers for detecting DoS attacks in Internet of Things (IoT) environments. Six machine learning classifiers were evaluated using the UNSW-NB15 dataset, namely, XGBoost, DT, Gaussian NB, RF, Logistic Regression (LR), and SVM. First, the dataset was processed with label encoding and normalization and then divided into a training set and a testing set. The results of the experiment indicated that the highest accuracy (99.48%), the highest precision (99.21%) and the highest sensitivity (99.69%) was obtained by the RF classifier. In addition, the Gaussian NB classifier achieved the lowest performance compared to the other models. The results demonstrated the suitability of ensemble learning methods, particularly RF, for IDS.

Authors in [17] proposed a novel hybrid deep learning algorithm to forecast DoS and DDoS attacks in the network security domain. The proposed model, CNN-LSTM-XGBoost, combines CNN for spatial feature extraction, LSTM for temporal dependence, and XGBoost for the final classification. After the preprocessing steps, such as null value removal, dealing with class imbalance, and feature selection based on correlation, the system was tested on three popular datasets: CICIDS001, CICIDS2017 and CICIDS2018. The evaluation results showed higher performance compared to existing

models, with an accuracy of 98.3% on CICIDS001, 99.2% on CICIDS2017, and 99.3% on the CICIDS2018 dataset.

Authors in [18] presented a novel deep learning technique for detecting various types of attacks using the NSL-KDD dataset. They proposed a Fast Hyper Deep Learning (FHDL) model that achieved 99% accuracy and perfect recall and F1-scores of 100%. In addition, the authors combined CNN with PCA and SVD to address the impact of redundant features. The preprocessing methodologies included normalization, label encoding, and feature selection to effectively enhance the model's performance. The proposed deep learning model, comprising 27 layers, including various convolutional and dense layers, successfully learned local and global patterns in network intrusion. While acknowledging the challenges posed by computational complexity and dataset biases to the efficiency of deep-learning based IDSs, the study demonstrates their efficiency.

In this paper, we introduce a customized deep learning model for the effective detection of DoS attacks in network traffic data. Unlike traditional machine learning approaches that rely on static feature extraction and shallow classifiers, the proposed model utilizes LSTM layers to capture temporal dependencies and sequential patterns inherent in network flows. The model architecture comprises multiple LSTM layers, which are designed to capture temporal dependencies in the data, followed by fully connected dense layers that perform the final classification. Prior to training, the input network data were preprocessed through feature normalization and reshaped into temporal sequences compatible with LSTM input requirements. The model was trained using a labeled dataset consisting of both normal and DoS traffic, allowing it to learn patterns that differentiate malicious behavior from legitimate network activity.

Our approach introduces several novel aspects. First, we leverage the sequential learning capabilities of LSTMs to capture time-based dependencies in network traffic—a frequently overlooked aspect in previous studies that tend to treat traffic data as independent samples. This approach is especially effective in detecting slow DoS attacks that unfold over time. Second, the model architecture was specifically designed and fine-tuned for the task of DoS detection, with careful optimization of hyperparameters such as the number of LSTM units, dropout rates, batch size, and learning rate. Finally, our focus was on achieving a performance balance that minimizes false positives while maintaining high classification accuracy, addressing one of the most persistent challenges in DoS detection systems.

III. PROPOSED METHODOLOGY

To address the challenges associated with detecting DoS attacks, we propose a comprehensive methodology designed to detect abnormal behavior with a high detection rate. The proposed methodology is outlined in Figure 1.

The proposed methodology begins with the NSL-KDD dataset, a widely recognized benchmark for evaluating network IDSs. The dataset undergoes preprocessing and normalization to prepare it for analysis. Preprocessing involves the encoding of categorical features and feature scaling. Normalization ensures that all features are on the same scale, a crucial step in improving the model's convergence. After preprocessing, the dataset is divided into two sets: training and testing. The training set is used for training the model, whereas testing set is reserved for evaluating the model's performance to ensure its ability to generalize to unseen data. In model evaluation, the trained LSTM is tested using the testing set, and evaluation metrics such as accuracy, precision, recall, and F1-score are computed to assess the model's performance.

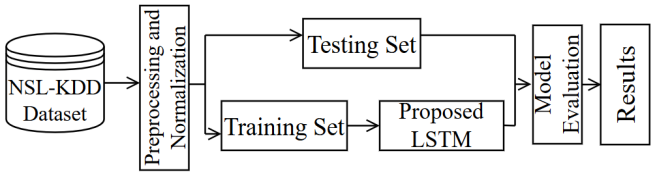

Fig. 1. Proposed methodology for detecting DoS attacks.

A. Dataset Description

The NSL-KDD [19] dataset is an enhanced version of the KDD Cup 1999 dataset, which is widely used for research in networks IDSs [20]. The KDD Cup 1999 dataset, while popular, exhibited several limitations, including a large amount of redundant and duplicate records, which could bias the performance of machine learning models. The NSL-KDD dataset was created to address these issues by eliminating redundant records and balancing the distribution of data across different classes, making it a more refined and reliable resource for evaluating intrusion detection algorithms [21]. The dataset contains 41 features and 148,517 samples that describe network connections. These features capture various aspects of network traffic, such as duration of the connection, protocol type, number of bytes transferred, and flag indicators. The dataset includes both normal and attack traffic, with attacks classified into four main categories: DoS, Remote to Local (R2L), User to Root (U2R), and Probe. Each record in the dataset is labeled as either normal or as a specific type of attack, making it suitable for supervised learning tasks. For this particular research, two subsets from the NSL-KDD dataset were used, which are NSL-KDD-Train and NSL-KDD-Test+. Table I describes all the features in the NSL-KDD dataset. These features are divided into three categorical attributes and the rest as numeric.

TABLE I. NSL-KDD DATASET FEATURE TYPES

Features	Type
F1, F5-F41	Numeric
F2-F4	Categorical

B. Preprocessing

The NSL-KDD dataset includes various attacks. In this study, the primary focus is on detecting DoS attacks within the NSL-KDD dataset. To simplify the classification task and emphasize the detection of DoS attacks, all other types of attacks (e.g., Probe, R2L, and U2R) are grouped and labeled as normal. This binary classification approach ensures that the model's performance is optimized for distinguishing the DoS attacks. Table II presents the distribution of the samples.

TABLE II. NSL-KDD DATASET ATTACK TYPES AND DISTRIBUTION

Category	Attack	Count	Total
Normal	Normal	77,054	95,132
	R2L	3,882	
	U2R	119	
	Probing (information gathering)	14,077	
DoS	DoS	53,385	53,385
Total			148,517

Standard Scaler is used for standardizing the input data of the dataset [22]. It works by normalizing the feature distributions so that each feature has a mean of 0 and a standard deviation of 1. This process ensures that all the attributes are on the same scale and contribute equally to the model, so that one attribute cannot dominate the learning process due to its larger scale. Normalizing the data in this way increases the convergence speed and efficiency [23]. The formula for standardization is:

$$X_{scaled}\frac{X-\mu}{\sigma} \quad (1)$$

where X is the original feature value, μ is the mean of the feature, and σ is the standard deviation.

Subsequently, the preprocessed data undergo one-hot encoding of features, including protocol type, service, and flag. One-hot encoding is applied to transform raw categorical data into binary vectors. Each category is encoded as a vector, where exactly one of its elements is 1, and the rest are 0. This is the usual procedure for encoding categorical data in cases where categories do not represent an ordinal relationship [24].

C. Proposed Long Short-Term Memory Model

Among all the variants of RNN, LSTM has become the standard for modeling sequential data with long-range dependencies. LSTM has feedforward connections and looping feedback connections, enabling the retention of information over a long period of time. This model has a unique memory cell where information can still be kept for quite some time and which forms the backbone of this model [25]. One of LSTMs' most notable aspects is that it uses neural network layers called gates to control information flow through the network. In fact, LSTM cell has three gates, forget, input, and output gates. Information that is vitally important is stored in the cell unit. The forget gate is used to filter the irrelevant data from the previous time step and passes only the relevant one. The input gate will selectively store relevant information from the current input, eliminating the unnecessary data. For instance, the cell state determines which information should be outputted from the output gate. In an LSTM, the values of status inside the cell can be modified in a controlled way through gate mechanisms [26]. The formulas of LSTM are as follows:

$$f_t = \sigma(W_f \cdot [a_{t-1}, x_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [a_{t-1}, x_t] + b_i) \quad (3)$$

$$g_t = tanh(W_c \cdot [a_{t-1}, x_t] + b_c) \quad (4)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot g_t \quad (5)$$

$$o_t = \sigma(W_o \cdot [a_{t-1}, x_t] + b_o) \quad (6)$$

$$a_t = o_t \cdot \tanh(C_t) \quad (7)$$

The forget gate output f_t specifies which parts of the former cell state C_{t-1} should be forgotten. Here, W_f is the weight matrix of the forget gate, a_{t-1} is the former hidden state, x_t is the current input, b_f is the bias for the forget gate, and σ is the function of sigmoid activation that gives values between 0 and 1. The input gate manages the addition of new information to the cell state and consists of the input gate activation i_t and the candidate cell state g_t, where g_t contains new information that is modulated by i_t to regulate the amount incorporated to the cell state C_t. The updated cell state C_t results from merging the forget and input gates, enabling the LSTM to selectively preserve past information C_{t-1} and integrate new data as required. The output gate determines the next hidden state a_t, which also serves as the LSTM cell output at the current time step o_t [27]. Figure 2 illustrates the functionality of the LSTM cell.

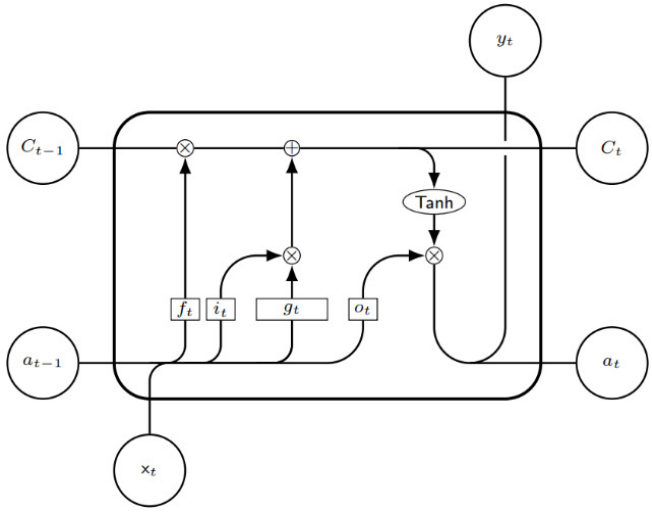

Fig. 2. LSTM cell functionality.

LSTM networks are designed to manage and retain long-term dependencies, making them well-suited for sequential data tasks such as natural language processing, speech recognition and IDSs. In IDSs, LSTM can detect intrusions, even those that are unknown or unexpected. There are several advantages of using LSTM compared to other algorithms for IDSs. First, the capability of learning long-term dependencies is critical for intrusion detection, spanning over many packets or events. Second, LSTM is more robust against noise, which is essential in real-world scenarios of IDSs. Lastly, LSTM is more efficient to train, making it ideal for IDSs that require training on large datasets [28].

Figure 3 illustrates the proposed LSTM model architecture. The structure is composed of multiple layers with integrated dropout layers for regularization, making it well-suited for binary classification tasks. The structure is designed to process sequential data effectively while reducing overfitting through dropout mechanisms. The model has an input layer as the entry point, so the data are input as sequential data that represent network traffic instances. Each instance has several features

values which describe a network behavior. Since LSTM networks are designed specifically for sequential dependencies, this layer is necessary for preparing the data so that they can be used for the rest of the network. The model consists of three stacked LSTM layers that process the input data at the core. This stacked architecture allows the network to learn high-order temporal patterns through the hierarchical refinement of information from previous layers. This architecture differs from the plain simple LSTM model in that dropout layers are inserted after the two first LSTM layers. To reduce overfitting and enhance the model's generalizability, dropout is employed during training to randomly deactivate a fraction of neurons.

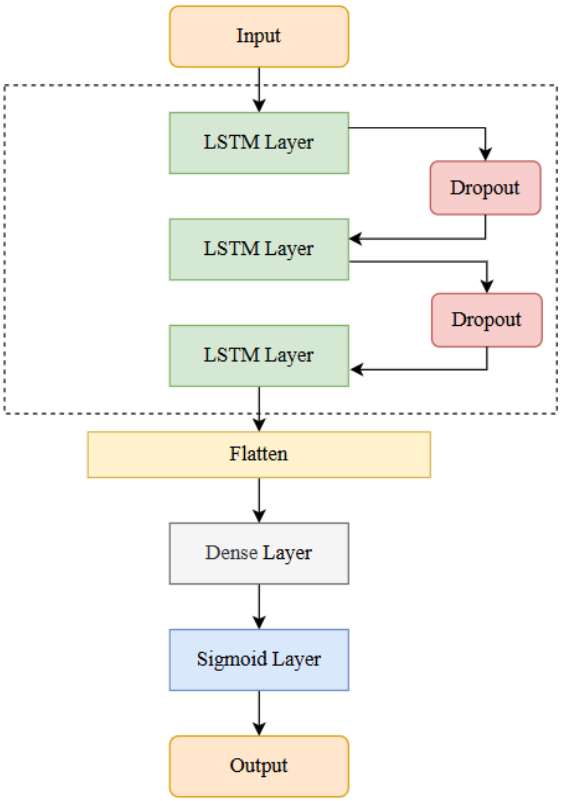

Fig. 3. The proposed LSTM model architecture.

After the LSTM layers, a flatten layer is introduced to flatten the multi-dimensional output of the LSTMs to a one-dimensional vector. This is needed to make the transition from LSTM layers to the subsequent fully connected layers in a fully connected way. Following this, the extracted features are processed by the dense layer (or fully connected layer), which learns higher level representations to assist differentiate between classes. This layer enables the model to generalize and refine information before making a last prediction. Then, a dense layer with a sigmoid activation is applied. This is used for the binary, as the sigmoid function maps the result to a probability value between 0 and 1 that describes the probability of the input belonging to DoS or a normal class. Finally, the output layer provides the classification result based on this probability. If the probability closer to 1, then the input is classified as DoS attack. Conversely, if the probability is closer to 0, then the input is classified as normal traffic. This architecture is expected to effectively leverages LSTMs to analyze network traffic patterns, enabling the detection of anomalies and cyber threats with high accuracy.

IV. RESULTS AND DISCUSSION

A. Evaluation Metrics

This section presents the evaluation metrics utilized to assess the proposed model on the NSL-KDD dataset. The experiment is conducted using Python with the Scikit-learn and the Keras libraries, and the evaluation is performed using well-established metrics, including accuracy, precision, recall, and F1-score. This approach ensures that the model's predictive capability is comprehensively judged. The calculation of metrics considers the values obtained from True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). In this context, TP signifies the traffic that is successfully predicted as an attack, whereas TN refers to normal traffic that is being predicated successfully as normal. In contrast, FP represents the normal traffic which has been wrongly predicated as an attack, whereas FN refers to the attack that has been incorrectly predicted as normal. The metrics are calculated according to the following equations:

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (8)$$

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

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (10)$$

$$\text{F1} - \text{score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision}+\text{Recall})} \quad (11)$$

B. Model Settings

To assess its effectiveness, the proposed LSTM is compiled with the Adam optimizer and loss binary cross-entropy. This architecture is well-suited for binary classification tasks, allowing the model to learn complex patterns from sequential data effectively. In this paper, the Adam optimizer is chosen due to its learning rate properties, which enhance convergence speed and stability, especially in deep neural networks. By combining the Adam optimizer with the binary cross-entropy loss, we ensure that the model adjusts its weights and biases efficiently to minimize any errors throughout training. The dropout value is set to 0.2, which indicates that 20% of the neurons in the specified layer will be randomly set to zero (deactivated) during each training iteration. This helps prevent overfitting by ensuring that the model does not become overly reliant on specific neurons, forcing it to learn more robust and generalized patterns.

The model was trained on the dataset using a total of 25 epochs, with an extra-large batch size of 32 to enhance memory efficiency and to allow the model to learn from sizable data chunks in each update cycle. To monitor performance and prevent overfitting, 20% of the training data was set aside as a validation split. During each epoch, the model is exposed to a considerable number of sequences, thereby enabling it to learn wide temporal dependencies within the data. The experiment trains the model on a substantial number of epochs and a large

batch size, whereas the split is incorporated to ensure that the learned patterns generalize well. This will ensure that the LSTM layers have sufficient data to function optimally, with a suitable balance of hyperparameters to provide an effective and smooth training. This training will yield insight into the model's predictive accuracy and stability across iterations.

C. Results

Figure 4 illustrates the training and validation accuracy of the proposed model, which represents the proportion of correctly classified cases out of the total cases. The model demonstrates a maximum level of accuracy of 99.92% in both the training set and the validation set. The overall average of the accuracy on the training and validation sets is 99.83% and 99.86%, respectively.

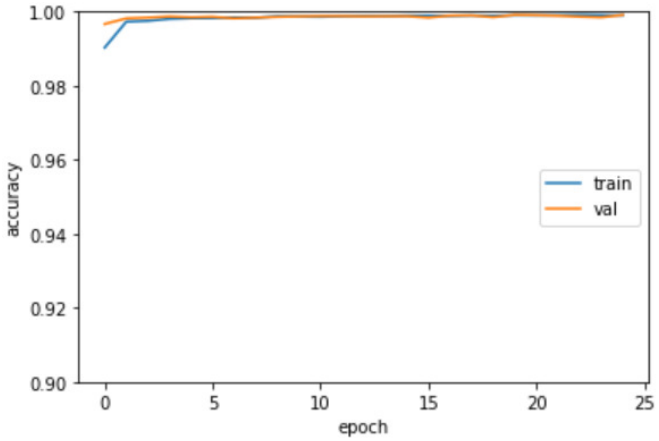

Fig. 4. Training and validation accuracy of the proposed LSTM model.

Table III presents the evaluation results of the proposed LSTM model across training, validation, and testing phases, highlighting four key performance metrics: accuracy, precision, recall, and F1-score. The model demonstrates consistently high accuracy, achieving approximately 99.83% and 99.86% during both training and validation, whereas testing accuracy reaches 99.92%, indicating strong generalization to unseen data. The precision values achieved during training and validation phases were 99.89% and 99.91%, respectively, with a notable increase to 99.93% in the testing phase. The recall values achieved were 99.84% and 99.87% for the training and validation phases, respectively. For the testing phase, the recall value was 99.94%. The F1-score, a metric that balances precision and recall, was 99.86% for the training phase, 99.89% for the validation phase, and reached a peak of 99.94% in the testing phase. These results highlight the robustness of the LSTM model, with a reliable performance during all phases, and an exceptional performance on the test set. This finding suggests that there is minimal overfitting and a high predictive capability.

Table IV presents a comparative analysis of various machine learning and deep learning models used for DoS and DDoS attack detection. The models were evaluated based on the accuracy, dataset, and methodology. The proposed model demonstrates superior performance, with accuracy rates of 99.92%, outperforming all previously reported models in the table. The highest accuracy achieved by prior models is 99.89%, achieved by authors in [14], using DT, which is not a deep learning method. This places the proposed model ahead by a notable margin, highlighting its effectiveness. Additionally, although other models such as LSTM-AE [15] (99.53%) and LSTM [13] (90.59%) also show commendable results, they do not match the accuracy of the proposed approach.

TABLE III. EVALUATION RESULTS OF THE PROPOSED LSTM MODEL

Set	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Training	99.83	99.89	99.84	99.86
Validation	99.86	99.91	99.87	99.89
Testing	99.92	99.93	99.94	99.94

TABLE IV. COMPARISON WITH OTHER MODELS

Reference	Accuracy (%)	Dataset	Method
[13]	94.28 90.59	UNSW-NB15 NSL-KDD	LSTM
[14]	99.89	NSL-KDD	DT
[15]	99.53	NSL-KDD	LSTM-AE
[16]	99.48	UNSW-NB15	RF
[17]	98.3 99.2 99.3	CICIDS-001 CIC-IDS2017 CIC-IDS2018	CNN-LSTM-XGBoost
[18]	99.0	NSL-KDD	FHDL
Proposed model	99.92	NSL-KDD	LSTM

The proposed model's utilization of sequential learning through LSTM provides a distinct advantage, contributing to the improved accuracy. Furthermore, the proposed model is benchmarked on the NSL-KDD dataset, ensuring fair and consistent comparisons with the majority of prior works. These results suggest that deep learning models, particularly LSTM, offer significant improvements in identifying cyber threats, necessitating further investigation in real-world scenarios.

V. CONCLUSION

This paper proposes a customized (Long Short-Term Memory) LSTM-based model for detecting Denial of Service (DoS) attacks using the NSL-KDD dataset. The model leverages LSTM's strength in capturing temporal dependencies within sequential data to improve detection performance. A comparison of the proposed model with existing approaches revealed its superior performance in terms of accuracy, precision, recall, and F1-score, demonstrating its effectiveness in identifying DoS attacks and reducing false positives.

A key contribution of this study lies in its focused optimization of the LSTM architecture for DoS detection, which sets it apart from traditional machine learning models and generic deep learning approaches that often overlook the sequential nature of network traffic data. Furthermore, when benchmarked against previous works using the same dataset, our model outperforms many state-of-the-art methods, showcasing its robustness and practical potential in Intrusion Detection Systems (IDSs). The novelty of this work also lies in its hybrid design perspective, laying the groundwork for future

integration with ensemble or boosting techniques to further refine detection accuracy. In future work, the model will be extended to support multi-class attack detection, incorporate additional deep learning architectures such as Bidirectional LSTM (BiLSTM) and Convolutional Neural Network (CNN), and be evaluated on real-time traffic data to enhance its generalizability and deployment readiness.

REFERENCES

[1] K. M. Canpolat and I. F. Kilincer, "Boosting Based IDS System for Local Network Intrusions," in *2024 8th International Artificial Intelligence and Data Processing Symposium*, Malatya, Turkiye, 2024, pp. 1–6, https://doi.org/10.1109/IDAP64064.2024.10710953.

[2] S. A. Ahmed, E. H. Khalifa, M. Nawaz, F. A. Abdalla, and A. F. A. Mahmoud, "Enhancing Cloud Data Center Security through Deep Learning: A Comparative Analysis of RNN, CNN, and LSTM Models for Anomaly and Intrusion Detection," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 20071–20076, Feb. 2025, https://doi.org/10.48084/etasr.9445.

[3] A. Nuhu, A. F. M. Raffei, M. F. A. Razak, and A. Ahmad, "Distributed Denial of Service Attack Detection in IoT Networks using Deep Learning and Feature Fusion: A Review," *Mesopotamian Journal of CyberSecurity*, vol. 4, no. 1, pp. 47–70, Apr. 2024, https://doi.org/10.58496/MJCS/2024/004.

[4] N. U. Ain, M. Sardaraz, M. Tahir, M. W. Abo Elsoud, and A. Alourani, "Securing IoT Networks Against DDoS Attacks: A Hybrid Deep Learning Approach," *Sensors*, vol. 25, no. 5, Mar. 2025, Art. no. 1346, https://doi.org/10.3390/s25051346.

[5] M. B. Anley, A. Genovese, D. Agostinello, and V. Piuri, "Robust DDoS attack detection with adaptive transfer learning," *Computers & Security*, vol. 144, Sep. 2024, Art. no. 103962, https://doi.org/10.1016/j.cose.2024.103962.

[6] M. A. I. Mallick and R. Nath, "Navigating the Cyber security Landscape: A Comprehensive Review of Cyber-Attacks, Emerging Trends, and Recent Developments," *World Scientific News*, vol. 190, no. 1, pp. 1–69, Jan. 2024.

[7] L. Diana, P. Dini, and D. Paolini, "Overview on Intrusion Detection Systems for Computers Networking Security," *Computers*, vol. 14, no. 3, Mar. 2025, Art. no. 87, https://doi.org/10.3390/computers14030087.

[8] Q. O. Ahmed, "Machine Learning for Intrusion Detection in Cloud Environments: A Comparative Study," *Journal of Artificial Intelligence General science*, vol. 6, no. 1, pp. 550–563, Dec. 2024, https://doi.org/10.60087/jaigs.v6i1.287.

[9] Y. Alraba'nah and W. Toghuj, "A deep learning based architecture for malaria parasite detection," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 292–299, Feb. 2024, https://doi.org/10.11591/eei.v13i1.5485.

[10] R. H. Altaie and H. K. Hoomod, "An Intrusion Detection System using a Hybrid Lightweight Deep Learning Algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16740–16743, Oct. 2024, https://doi.org/10.48084/etasr.7657.

[11] R. Jablaoui and N. Liouane, "Network security based combined CNN-RNN models for IoT intrusion detection system," *Peer-to-Peer Networking and Applications*, vol. 18, no. 3, Mar. 2025, Art. no. 129, https://doi.org/10.1007/s12083-025-01944-7.

[12] Z. S. Dhahir, "A Hybrid Approach for Efficient DDoS Detection in Network Traffic Using CBLOF-Based Feature Engineering and XGBoost," *Journal of Future Artificial Intelligence and Technologies*, vol. 1, no. 2, pp. 174–190, Sep. 2024, https://doi.org/10.62411/faith.2024-33.

[13] C. C. Ugwu, O. O. Obe, O. S. Popoọla, and A. O. Adetunmbi, "A Distributed Denial of Service Attack Detection System using Long Short Term Memory with Singular Value Decomposition," in *2020 IEEE 2nd International Conference on Cyberspac (CYBER NIGERIA)*, Abuja, Nigeria, 2021, pp. 112–118, https://doi.org/10.1109/CYBERNIGERIA51635.2021.9428870.

[14] M. M. Abualhaj, A. A. Abu-Shareha, M. O. Hiari, Y. Alrabanah, M. Al-Zyoud, and M. A. Alsharaiah, "A Paradigm for DoS Attack Disclosure using Machine Learning Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 3, pp. 192–200, 2022, https://doi.org/10.14569/IJACSA.2022.0130325.

[15] S. Sumathi, R. Rajesh, and S. Lim, "Recurrent and Deep Learning Neural Network Models for DDoS Attack Detection," *Journal of Sensors*, vol. 2022, no. 1, Sep. 2022, Art. no. 8530312, https://doi.org/10.1155/2022/8530312.

[16] O. Almomani, A. Alsaaidah, A. A. A. Shareha, A. Alzaqebah, and M. Almomani, "Performance Evaluation of Machine Learning Classifiers for Predicting Denial-of-Service Attack in Internet of Things," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 1, pp. 263–271, 2024, https://doi.org/10.14569/IJACSA.2024.0150125.

[17] A. F. Al-zubidi, A. K. Farhan, and S. M. Towfek, "Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model," *Journal of Intelligent Systems*, vol. 33, no. 1, Jan. 2024, Art. no. 20230195, https://doi.org/10.1515/jisys-2023-0195.

[18] H. M. S. Saleeh, H. Marouane, and A. Fakhfakh, "A Novel Deep Learning Approach for Detecting Types of Attacks in the NSL-KDD Dataset," *Babylonian Journal of Networking*, vol. 2024, pp. 171–181, Sep. 2024, https://doi.org/10.58496/BJN/2024/017.

[19] G. Mohi-ud-din, "NSL-KDD." IEEE DataPort, Dec. 29, 2018, https://doi.org/10.21227/425A-3E55.

[20] A. A. Abu-Shareha and M. M. Abualhaj, "Improving Intrusion Detection System Using Feature Weighting," in *Soft Computing and Its Engineering Applications: 6th International Conference, icSoftComp 2024*, Bangkok, Thailand, 2024, pp. 147–160, https://doi.org/10.1007/978-3-031-88042-1_12.

[21] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and J. S. Alqurni, "CNN Channel Attention Intrusion Detection System Using NSL-KDD Dataset," *Computers, Materials & Continua*, vol. 79, no. 3, pp. 4319–4347, Jun. 2024, https://doi.org/10.32604/cmc.2024.050586.

[22] V. V. Rama Rao M, A. Rapaka, M. Prasad, R. R. PBV, P. T. Satyanarayana Murty, and K. S. Pokkuluri, "Enhancing Network Security: Leveraging Machine Learning for Intrusion Detection," *Journal of Electrical Systems*, vol. 20, no. 2, pp. 1555–1562, Apr. 2024, https://doi.org/10.52783/jes.1460.

[23] M. A. Talukder, M. Khalid, and N. Sultana, "A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction," *Scientific Reports*, vol. 15, no. 1, Feb. 2025, Art. no. 4617, https://doi.org/10.1038/s41598-025-87028-1.

[24] M. Alenazi and S. Mishra, "Cyberatttack Detection and Classification in IIoT systems using XGBoost and Gaussian Naïve Bayes: A Comparative Study," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 15074–15082, Aug. 2024, https://doi.org/10.48084/etasr.7664.

[25] S. S. Bamber, A. V. R. Katkuri, S. Sharma, and M. Angurala, "A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system," *Computers & Security*, vol. 148, Jan. 2025, Art. no. 104146, https://doi.org/10.1016/j.cose.2024.104146.

[26] I. M. Elezmazy and N. N. Mostafa, "Enhanced Network Security using LSTM-Based Autoencoder Models," *Artificial Intelligence in Cybersecurity*, vol. 1, pp. 60–69, Jun. 2024, https://doi.org/10.61356/j.aics.2024.1315.

[27] H. Yadav and A. Thakkar, "NOA-LSTM: An efficient LSTM cell architecture for time series forecasting," *Expert Systems with Applications*, vol. 238, no. F, Mar. 2024, Art. no. 122333, https://doi.org/10.1016/j.eswa.2023.122333.

[28] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," *Information*, vol. 15, no. 9, Sep. 2024, Art. no. 517, https://doi.org/10.3390/info15090517.