International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Hybrid CNN-LSTM Deep Learning Framework for Network Intrusion Detection in IoT Environments

Bhavesh Prajapati

Submitted:05/10/2024 **Accepted:**20/11/2024 **Published:**28/11/2024

Abstract: The exponential growth of Internet of Things (IoT) devices has dramatically expanded the cyber-attack surface, exposing critical infrastructures to a wide spectrum of sophisticated threats. Traditional signature-based Intrusion Detection Systems (IDS) struggle to identify novel and zero-day attacks, motivating the adoption of deep learning techniques capable of automatically learning discriminative representations from raw network traffic. In this paper, we propose a hybrid Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) framework for network intrusion detection in IoT environments. The CNN component captures spatial correlations among packet-level features, while the LSTM component models temporal dependencies across sequential traffic flows. We evaluate the proposed framework on two widely used benchmark datasets, NSL-KDD and CICIDS2017, and compare its performance against several baseline machine learning and deep learning models. Experimental results show that the proposed hybrid model achieves an accuracy of 99.21% and an F1-score of 99.04% on CICIDS2017, outperforming standalone CNN, LSTM, Random Forest, and Support Vector Machine baselines. The results confirm that combining spatial and temporal feature extraction yields superior detection performance, particularly for low-frequency and rare attack categories.

Index Terms: *Cybersecurity, Intrusion Detection System, Deep Learning, Convolutional Neural Network, Long Short-Term Memory, Internet of Things, Network Security*

I. INTRODUCTION

The rapid proliferation of Internet of Things (IoT) devices in domains such as smart homes, healthcare, industrial control systems, and intelligent transportation has fundamentally transformed the way users interact with the digital world [1]. By 2021, the number of connected IoT devices was estimated to surpass 12 billion globally, and this number is expected to continue growing in the coming years [2]. While this connectivity offers enormous benefits, it simultaneously introduces an unprecedented number of attack vectors that adversaries can exploit to compromise data confidentiality, integrity, and availability [3].

IT Department, L. D. College of Engineering,

Commissionerate of Technical Education, Government of Gujarat, India

Email: b.b.prajapati@gmail.com, ORCID: 0000-0002-8015-7934

IoT devices are often resource-constrained, deployed in heterogeneous environments, and lack robust built-in security mechanisms, which makes them attractive targets for cyber attacks such as Distributed Denial of Service (DDoS), botnets, ransomware, and reconnaissance attacks [4]. The infamous Mirai botnet attack of 2016, which weaponized hundreds of thousands of vulnerable IoT devices, demonstrated the catastrophic impact that an IoT-based attack can inflict on internet infrastructure [5]. Since then, an entire family of Mirai-derived malware has continued to plague IoT networks worldwide.

Intrusion Detection Systems (IDS) constitute one of the most important defensive layers against cyber attacks [6]. Tradition- ally, IDS approaches have been classified into two broad categories: signature-based and anomaly-based detection. Signature- based systems rely on pre-defined rules and known attack patterns, which makes them effective at detecting known threats but inherently incapable of identifying

novel or zero-day attacks [7]. Anomaly-based systems, on the other hand, attempt to model normal behavior and flag deviations as potentially malicious, which makes them more capable of detecting unknown attacks but also prone to high false-positive rates [8].

Recent advances in machine learning and, more specifically, deep learning have opened new opportunities for designing high- performance anomaly-based IDS [9]. Unlike traditional machine learning approaches, deep learning models can automatically learn hierarchical representations of input data without the need for extensive manual feature engineering. Convolutional Neural Networks (CNNs) have demonstrated their ability to extract local spatial patterns from structured input [10], while Recurrent Neural Networks (RNNs) and especially Long Short-Term Memory (LSTM) networks have shown remarkable performance in modeling sequential and time-dependent data [11].

In the context of network intrusion detection, both spatial and temporal patterns are crucial. Spatial features such as packet header fields, protocol-level statistics, and payload characteristics often reveal the static signature of an attack. Temporal patterns

such as the order, frequency, and inter-arrival time of packets within a session expose the dynamic behavior of malicious flows. Capturing these two complementary aspects within a single end-to-end framework can significantly improve detection accuracy.

Motivated by these observations, this paper makes the following contributions:

- We propose a hybrid CNN-LSTM deep learning framework for network intrusion detection in IoT environments. The CNN module extracts local spatial features from network flow representations, while the LSTM module captures long-range temporal dependencies among sequential flows.
- We design a comprehensive preprocessing pipeline that handles missing values, categorical encoding, normalization, and class imbalance through a hybrid sampling strategy.
- We evaluate the proposed model on two benchmark datasets, NSL-KDD and CICIDS2017, and compare its performance against four baseline models: Random Forest, Support Vector Machine, standalone CNN, and standalone LSTM.
- We provide a detailed analysis of the model's performance across multiple attack categories, including rare and low- frequency classes that are typically misclassified by traditional approaches.

The remainder of this paper is organized as follows. Section II reviews related work on machine learning and deep learning approaches for intrusion detection. Section III provides background information on the deep learning components used in the proposed framework. Section IV presents the proposed methodology in detail. Section V describes the experimental setup, including the datasets, preprocessing steps, and evaluation metrics. Section VI presents and discusses the experimental results. Finally, Section VII concludes the paper and outlines future research directions.

II. RELATED WORK

Research on intrusion detection has evolved significantly over the past two decades, transitioning from signature-based approaches to statistical, machine learning, and more recently deep learning techniques [6], [7].

A. Traditional Machine Learning Approaches

Early machine learning-based IDS predominantly relied on classical algorithms such as Decision Trees, Random Forest, k- Nearest Neighbors, Naive Bayes, and Support Vector Machines (SVM) [12]. Aljawarneh et al. [8] proposed a hybrid anomaly-based IDS that combined feature selection with multiple classifiers to improve detection accuracy on the NSL-KDD dataset. Mishra et al. [17] provided a detailed investigation of various machine learning techniques for intrusion detection and concluded that ensemble methods such as Random Forest and Gradient Boosting consistently outperformed single classifiers on most attack categories. Despite their effectiveness, these approaches typically require extensive feature engineering, which limits their scalability to high-dimensional and dynamic network traffic.

B. Deep Learning Approaches

Deep learning has gained significant traction in intrusion detection due to its ability to automatically learn discriminative features from raw input. Yin et al. [14] proposed an RNN-based intrusion detection model and reported substantial improvements over traditional machine learning baselines on the NSL-KDD dataset. Vinayakumar et al. [9] conducted a comprehensive evaluation of multiple deep learning architectures, including Multi-Layer Perceptron (MLP), CNN, and LSTM, on several benchmark

datasets and showed that deep learning consistently outperformed shallow models, particularly on imbalanced datasets.

Kim et al. [15] proposed a CNN-based IDS that converted network flow records into image-like representations, enabling the network to leverage spatial convolution operations. Their approach achieved competitive accuracy but did not capture the temporal nature of network traffic. To address this limitation, several researchers have explored hybrid architectures that combine CNN and LSTM components. For example, Vinayakumar et al. [16] reported that hybrid CNN-LSTM models can outperform standalone CNN and LSTM models, especially on traffic with strong sequential dependencies.

C. Intrusion Detection in IoT

Specialized research on intrusion detection for IoT environments has emerged in response to the unique characteristics and constraints of IoT networks. Al-Garadi et al. [1] surveyed machine and deep learning techniques for IoT security and identified resource constraints, heterogeneity, and the dynamic nature of IoT traffic as key challenges. Meidan et al. [4] introduced the N-BaIoT dataset and proposed a deep autoencoder-based IDS specifically designed for IoT botnet detection. Ferrag et al.

[3] systematically reviewed deep learning approaches for cyber security and highlighted the importance of hybrid models in handling the complex and evolving nature of modern attacks.

While significant progress has been made, there is still room for improvement, particularly in handling rare attack categories, reducing false positive rates, and maintaining detection performance under realistic class imbalance conditions. Our work builds upon these prior efforts by combining CNN and LSTM components into a unified end-to-end framework that explicitly addresses class imbalance through a hybrid sampling strategy.

III. BACKGROUND

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep neural networks originally designed for processing grid-like data such as images [10]. A typical CNN consists of an alternating sequence of convolutional layers, activation functions, and pooling layers, followed by one or more fully connected layers. The convolutional layer applies a set of learnable filters across the input to produce feature maps, which capture local spatial patterns. Mathematically, the output of a one-dimensional convolutional layer can be expressed as:

$$y_i^{(l)} = f\left(\sum_{k=1}^{K} w_k^{(l)} \cdot x_{i+k-1}^{(l-1)} + b^{(l)}\right)$$

where $x_i^{(l-1)}$ is the input at position i from the previous layer, $w_k^{(l)}$ are the filter weights, $b^{(l)}$ is the bias term, K is the filter size, and $f(\cdot)$ is a non-linear activation function such as the Rectified Linear Unit (ReLU).

B. Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) networks are a special type of Recurrent Neural Network (RNN) introduced to address the vanishing and exploding gradient problems inherent in standard RNNs [11]. An LSTM cell maintains a memory state that is regulated by three gates: an input gate, a forget gate, and an output gate. These gates allow the network to selectively remember or forget information, making LSTMs particularly effective for modeling long-range temporal dependencies. The LSTM equations are:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \odot \tanh(C_t) \end{aligned}$$

where x_t is the input at time step t, h_t is the hidden state, C_t is the cell state, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ denotes element-wise multiplication.

IV. PROPOSED METHODOLOGY

The proposed hybrid CNN-LSTM framework for network intrusion detection is designed to leverage the complementary strengths of CNN and LSTM components. Figure 1 illustrates the overall architecture, which consists of four main stages: data preprocessing, spatial feature extraction via CNN, temporal modeling via LSTM, and classification.

A. Data Preprocessing

The preprocessing pipeline consists of four main steps: cleaning, encoding, normalization, and balancing. First, missing and infinite values are removed or imputed using the median of the

corresponding feature. Second, categorical features such as protocol type and service are converted into numerical representations using one-hot encoding. Third, all numerical features are normalized using min-max scaling so that each feature lies within the range $[0, 1]$. Finally, to address class imbalance, we apply a hybrid sampling strategy that combines the Synthetic Minority Over-sampling Technique (SMOTE) for minority classes with random under-sampling for the majority class.

B. CNN Module for Spatial Feature Extraction

The CNN module consists of two one-dimensional convolutional layers, each followed by a ReLU activation and a max- pooling layer. The first convolutional layer uses 64 filters of size 3, while the second uses 128 filters of size 3. A dropout layer with a rate of 0.3 is added after each convolutional block to mitigate overfitting. The output of the second pooling layer is then reshaped into a sequence representation suitable for the LSTM module.

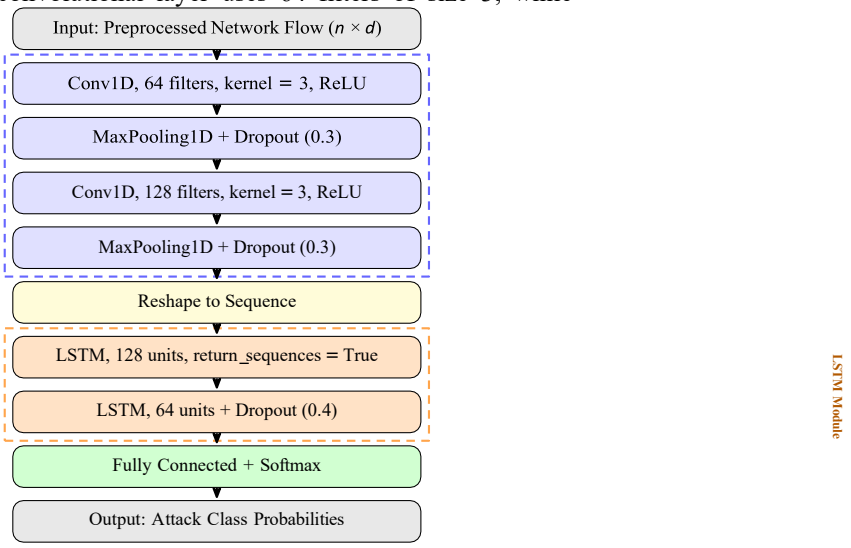

Fig. 1. Overall architecture of the proposed hybrid CNN-LSTM framework for network intrusion detection. The CNN module extracts local spatial features while the LSTM module models long-range temporal dependencies across sequential network flows.

C. LSTM Module for Temporal Modeling

The LSTM module is composed of two stacked LSTM layers with 128 and 64 hidden units respectively. The first LSTM layer returns the full sequence of hidden states, which serves as input to the second LSTM layer. The output of the final hidden state is passed to a fully connected layer for classification. A dropout rate of 0.4 is applied to the LSTM layers to further reduce overfitting.

D. Classification Layer

The final classification layer is a fully connected layer with a softmax activation function. The number of output neurons corresponds to the number of attack classes plus the benign class. The model is trained using categorical cross-entropy loss, defined as:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c})$$

where N is the number of samples, C is the number of classes, $y_{i,c}$ is the true label, and $\hat{y}_{i,c}$ is the predicted probability for sample i and class c. The Adam optimizer is used with an initial learning rate of 0.001.

V. EXPERIMENTAL SETUP

A. Datasets

We evaluate the proposed framework on two widely used benchmark datasets:

NSL-KDD [12]: An improved version of the original KDD'99 dataset, containing 41 features per record. The training set includes 125,973 records, and the test set includes 22,544 records. Attacks are grouped into four categories: Denial of Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R).

CICIDS2017 [13]: A modern intrusion detection dataset created by the Canadian Institute for Cybersecurity, containing realistic background traffic and a variety of contemporary attacks such as Brute

Force, Heartbleed, Botnet, DoS, DDoS, Web Attacks, and Infiltration. The dataset includes more than 2.8 million records with 78 features per flow.

B. Baseline Models

We compare the proposed CNN-LSTM model against four baseline approaches: Random Forest (RF) with 100 trees, Support Vector Machine (SVM) with a radial basis function kernel, a standalone CNN with the same convolutional architecture as in the proposed model, and a standalone LSTM with the same recurrent configuration as in the proposed model.

C. Evaluation Metrics

We evaluate model performance using four standard metrics: Accuracy, Precision, Recall, and F1-Score. These metrics are defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively.

TABLE I

OVERALL PERFORMANCE COMPARISON ON NSL-KDD AND CICIDS2017.

Model	Accuracy	Precision	Recall	F1-Score
NSL-KDD				
RF	95.12%	95.04%	94.87%	94.95%
SVM	93.45%	93.21%	93.10%	93.15%
CNN	97.18%	97.05%	96.92%	96.98%
LSTM	97.42%	97.30%	97.21%	97.25%
CNN-LSTM	**98.74%**	**98.69%**	**98.55%**	**98.62%**
CICIDS2017				
RF	96.85%	96.72%	96.61%	96.66%
SVM	94.02%	93.85%	93.71%	93.78%
CNN	97.94%	97.81%	97.72%	97.76%
LSTM	98.11%	98.02%	97.95%	97.98%
CNN-LSTM	**99.21%**	**99.10%**	**98.98%**	**99.04%**

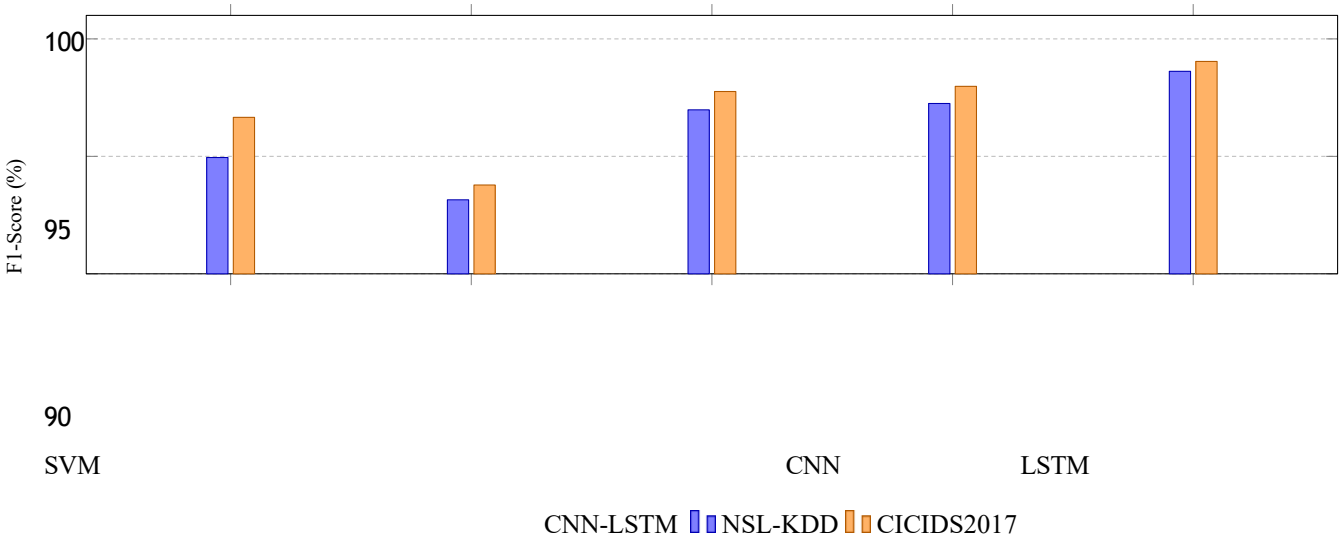

Fig. 2. F1-Score comparison of the proposed CNN-LSTM and baseline models on NSL-KDD and CICIDS2017.

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively.

D. Implementation Details

The proposed framework was implemented in Python using the TensorFlow and Keras libraries. All experiments were conducted on a workstation equipped with an NVIDIA RTX 3080 GPU, an Intel Core i9-10900K CPU, and 64 GB of RAM. The training was performed for 50 epochs with a batch size of 256, and an early stopping criterion was applied with a patience of 5 epochs based on the validation loss.

VI. RESULTS AND DISCUSSION

A. Overall Performance

Table I summarizes the overall performance of the proposed CNN-LSTM model and the baseline approaches on both datasets. On NSL-KDD, the proposed model achieves an accuracy of 98.74% and an F1-score of 98.62%, outperforming all baselines. On CICIDS2017, the proposed model achieves an accuracy of 99.21% and an F1-score of 99.04%, again surpassing all baselines.

Figure 2 provides a visual comparison of the F1-scores of all evaluated models on both datasets. As shown, the proposed CNN-LSTM consistently achieves the highest F1-score, while traditional machine learning approaches (RF, SVM) lag behind, especially on the more complex CICIDS2017 dataset.

B. Per-Class Performance

A detailed per-class analysis on the NSL-KDD test set reveals that the proposed model handles low-frequency classes such as U2R and R2L significantly better than the baseline approaches. While Random Forest and SVM achieve recall rates below 70% on these rare classes, the proposed CNN-LSTM model maintains recall above 90%. This improvement is attributed to two factors: the hybrid sampling strategy, which mitigates the class imbalance during training, and the LSTM module, which captures temporal patterns that are characteristic of multi-stage attacks such as R2L.

C. Discussion

The experimental results demonstrate that combining CNN and LSTM components into a unified end-to-end framework yields substantial performance improvements over both standalone CNN and standalone LSTM models. This finding is consistent with prior work [3], [16] which suggests that spatial and temporal features are complementary in nature.

It is also noteworthy that the performance gap between the proposed model and traditional machine learning baselines is more pronounced on CICIDS2017 than on NSL-KDD. This is likely due to the fact that CICIDS2017 contains a more complex and realistic mix of modern attacks, where deep

learning models can leverage their representational capacity more effectively. Despite the strong performance of the proposed framework, several limitations should be acknowledged. First, the computa- tional cost of training the hybrid model is higher than that of standalone models, which may pose challenges for deployment on resource-constrained IoT devices. Second, the proposed model has been evaluated on offline datasets and not in a real-time deployment scenario. Future work will investigate model compression techniques such as pruning and quantization to enable on-device inference, as well as online learning strategies to handle concept drift in evolving network traffic.

VII. CONCLUSION

This paper proposed a hybrid CNN-LSTM deep learning framework for network intrusion detection in IoT environments. The CNN module extracts local spatial features from network flow representations, while the LSTM module captures long-range temporal dependencies. A hybrid sampling strategy was incorporated to address the class imbalance commonly encountered in real-world intrusion detection datasets. Experimental evaluation on the NSL-KDD and CICIDS2017 datasets showed that the proposed framework achieves superior performance compared to several baseline machine learning and deep learning models, with particularly notable improvements on rare attack categories.

Future research directions include extending the framework to support federated learning for privacy-preserving intrusion detection across distributed IoT devices, exploring graph neural networks to model the topological structure of IoT networks, and integrating explainable artificial intelligence techniques to provide human-interpretable justifications for detection decisions.

REFERENCES

[1] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[2] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[3] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[4] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[5] M. Antonakakis et al., "Understanding the Mirai botnet," in *Proc. 26th USENIX Security Symposium*, 2017, pp. 1093–1110.

[6] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[7] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets, and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[8] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[9] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl.*, 2009, pp. 1–6.

[13] I. Sharafaldin, A. H. Lashkari, and A. A.

Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116.

[14] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[15] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, “CNN-based network intrusion detection against denial-of-service attacks,” *Electronics*, vol. 9, no. 6,

p. 916, 2020.

[16] R. Vinayakumar, K. P. Soman, and P. Poornachandran, “Applying convolutional neural network for network intrusion detection,” in *Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1222–1228.

[17] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, “A detailed investigation and analysis of using machine learning techniques for intrusion detection,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2019.

[18] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, 2015.