

A HYBRID DEEP LEARNING APPROACH FOR INTRUSION DETECTION IN NETWORK TRAFFIC USING CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS

1M Arslan Saleem Sandila, 2Sidra Sultan, 3Zaheer ul Hassan, 4Arshad Ali,

MS in Information Security, Department of Computer Science at Bahria University, Islamabad

arslansandila@gmail.com

MS in Electrical Engineering, Institute of Space Technology, Islamabad-sultan_sidrah@yahoo.com

MS in Computer Science, Department of Computer Science at Bahria University, Karachi

zaheerulhassank@outlook.com

MS in Information Security Department of Computer Science at Bahria University, Islamabad

Ltarshadali@gmail.com

Keywords

Hybrid Model, Network Traffic Analysis, Cybersecurity, NSL-KDD Dataset, Temporal Feature Extraction, Spatial Feature Learning

Article History

Received on 01 June 2025
Accepted on 25 June 2025
Published on 27 July 2025

Copyright @Author

M Arslan Saleem Sandila

Abstract

An increase in the role of digital communication and cloud-based infrastructures has greatly magnified the size and intelligence of cyber threats and continues to present problems for traditional Intrusion Detection Systems (IDS). Specifically, traditional IDS mechanisms usually based on either static signatures or shallow learning approaches are governed by limited parameters that do not provide the capacity to adapt or be robust with regards to the detection of new or obfuscated attacks. In order to overcome these drawbacks, we propose in this paper a hybrid deep learning-based intrusion detection framework that jointly combines the use of Convolutional Neural Networks (CNNs) to extract spatial features and the usage of Long Short-Term Memory (LSTM) networks to learn temporal features in a sequence of network traffic data. In the proposed model, preprocessed and normalized network flow features go through a pipeline and are inputted into several convolutional filters where they automatically learn high-dimensional, abstract spatial features. The extracted learned features are then fed into LSTM units sequentially, which is able to model the dynamic temporal behavior of network sessions and capture multi-stage or time-evolving attack patterns. It refines the architecture by employing categorical cross-entropy loss and optimized with the Adam optimizer and trained on benchmark NSL-KDD dataset. The hybrid CNN-LSTM model achieves a detection accuracy of 96.78%, with precision, recall, and F1-score all over 95%, and a comprehensive experimental validation shows that the proposed hybrid model significantly outperforms standalone CNN, LSTM and classical machine learning algorithms (SVM and Random Forest).

INTRODUCTION

The uncharted growth in cyberspace and the ubiquitous distribution of systems, IoT, and cloud-native architecture have created exponentially larger complexity and sizes of network traffic. However, this digital transformation has somehow provided the

opportunity for more sophisticated cyberattacks such as distributed denial of service (DDoS), reconnaissance, remote-to-local (R2L), and user-to-root (U2R) attacks which has been previously noted [1]. This adversarial behavior is polymorphic, stealthy, and temporally dispersed, thus cannot be easily detected by traditional Intrusion Detection System (IDS). Conventional IDS frameworks mostly utilize either signature-based or statistical anomaly detection techniques that involve manually defined patterns and thresholds. Due to the static way in which these systems operate, their inability to generalize, and their lack of detection capability against zero-day and evolving attack vectors, these systems are inherently limited [2]. When classical machine learning techniques are executed e.g. Random Forests, Support Vector Machines or Logistic Regression, they frequently rely on extensive feature engineering and do not have sufficient representational capacity to capture the high dimensional, nonlinear and temporal behaviours of network intrusions[3].

Deep Learning (DL) is a disruptive technology that has changed the paradigm in many fields driven by its ability to extract features automatically and learn features hierarchies. More specifically, Convolutional Neural

Networks (CNN) tend to capture spatial local patterns and localized statistical dependencies across features while the Recurrent Neural Networks (RNN)s, especially Long Short-Term Memory (LSTM) architectures are learned well for long-term temporal dependencies of data in sequence [4]. But each model comes with its own set of limitations: CNNs are not capable of modeling temporal dependencies, and RNNs can not learn low-level spatial feature representations from static inputs. In response to these issues, we present a single hybrid deep learning architecture that combines both 1D CNN and LSTM networks in the such a way as to form a single pipeline [5]. Such model aims to capture abundant spatiotemporal feature representations from normalized network flow data and to classify various intrusion types effectively. A CNN based component extracts each multidimensional feature space through a convolutional operation and learns a space-discriminative filter by applying weight sharing in the convolutional layers. The extracted features are then fed into the LSTM layer that models the temporal dynamics and session-to-session (packet-to-packet) dependence—foundational for identifying multi-stage or low-rate exploits [6].

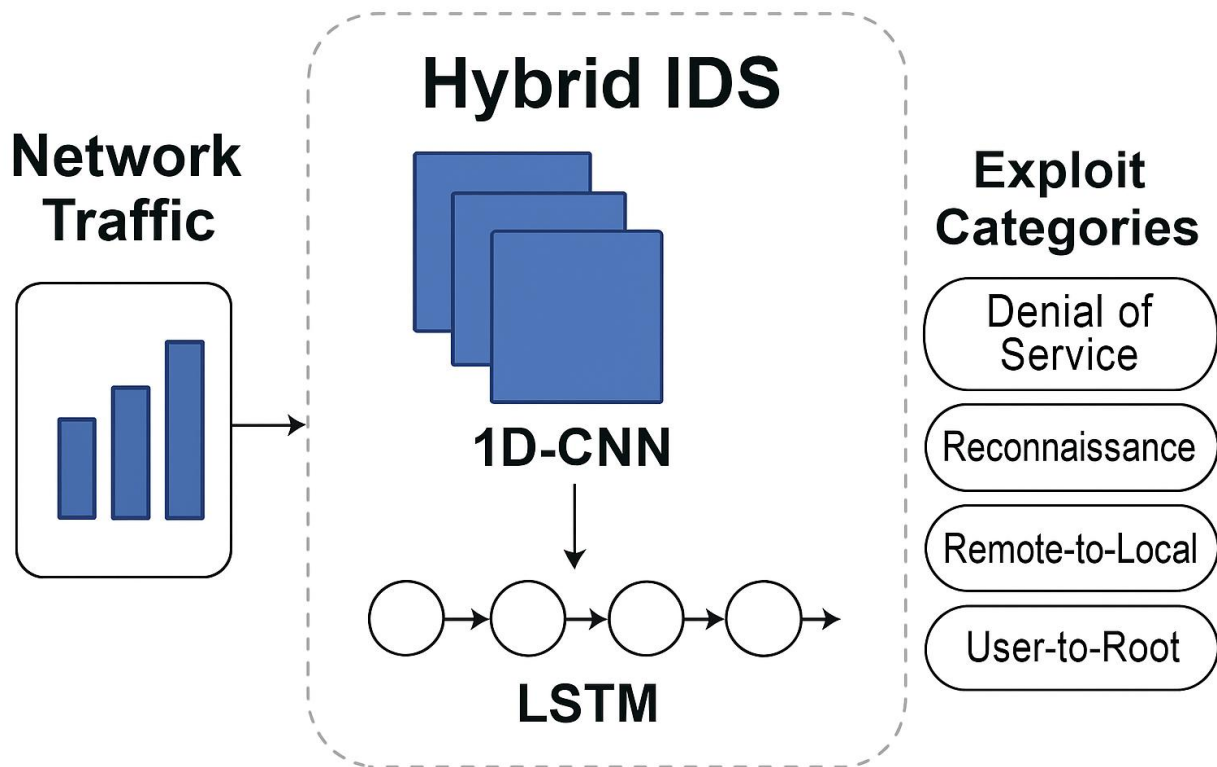


Fig 1.1 illustrates the architecture of a Hybrid Intrusion Detection System (IDS) that combines 1D-CNN and LSTM for analyzing network traffic. The raw network traffic is first processed by 1D-CNN layers, which extract spatial and local feature patterns from packet sequences. These extracted features are then passed to the LSTM network, which captures sequential dependencies and temporal relationships in the traffic data. Finally, the model classifies the traffic into different exploit categories such as Denial of Service, Reconnaissance, Remote-to-Local and User-to-Root attacks, enabling robust intrusion detection.

Mathematically, given an input matrix $X \in \mathbb{R}^{n \times m}$ where n is the number of features and m represents time-steps or temporal slices, the CNN performs feature map extraction via:

$$Y_i = \text{ReLU}\left(\sum_{j=1}^{K-1} W_j \cdot X_i + j + b\right)$$

where W_j are learnable filters of size K and b is the bias term. The output is subsequently fed into the LSTM, which recursively updates its hidden state h and cell state C using the gating mechanisms:

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

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

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot C_t$$

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = O_t \odot \tanh(C_t)$$

This dual mechanism enables the system to preserve information of previous characteristics and to react continuously to

novel inputs, thereby efficiently representing intricate network dynamics [7,8]. We conduct empirical evaluation of the proposed architecture over NSL-KDD dataset, which is a well known benchmark that resolves the redundancy and imbalance problems associated with earlier datasets including KDD-99. We perform extensive experiments to evaluate their corresponding classification accuracy, as well as attack specific precision, recall, F1-score in addition to confusion matrix and Receiver Operating Characteristic (ROC) analysis.

1.1 Research Contribution

- A novel CNN-LSTM hybrid model that fuses spatial and sequential learning paradigms to enhance intrusion detection efficacy across all attack classes.
- A deep feature extraction pipeline that autonomously learns hierarchical representations from raw traffic data without manual feature engineering.
- Robust evaluation on benchmark IDS datasets, demonstrating superior performance over standalone CNN, LSTM, and traditional machine learning models.
- Mathematical formulation and modular design, enabling reproducibility and real-time

deployment in intelligent network monitoring systems.

2. Literature Review

The growing complexity of modern cyber threats has driven researchers to adopt hybrid deep learning architectures that combine the strengths of convolutional and recurrent networks for intrusion detection. In this article [9] authors presented a scalable CNN-LSTM-based hybrid intrusion detection system capable of capturing spatial features through convolutional layers and sequential dependencies using LSTM, demonstrating superior accuracy and adaptability for large-scale traffic environments. Similarly in another article [10] employed CNNs for malicious traffic detection in computer networks, showing the effectiveness of deep spatial feature extraction in reducing false positives compared to traditional machine learning methods. Beyond intrusion detection, deep learning has also shown promise in modeling complex biological and computational systems. The authors [11] applied Graph Neural Networks (GNNs) to predict viral protein folding dynamics and thermodynamic stability, highlighting the broader applicability of advanced neural architectures in domains where structural and sequential dependencies play a crucial role an analogy that strengthens

the case for CNN-LSTM hybrids in cybersecurity.

Comparative evaluations of deep learning models for intrusion detection further confirm the dominance of hybrid architectures. This research paper [12] provided an extensive benchmarking study of CNN, RNN, LSTM, and hybrid configurations, concluding that integrated models consistently outperform standalone architectures in terms of accuracy, sensitivity, and false alarm reduction. Reinforced this view by proposing a hybrid deep learning IDS tailored for big data environments, where scalability and efficient feature learning are essential. Advancements in software-defined networking (SDN) and feature engineering have inspired refined hybrid models [13]. In this article [14] introduced CNN-BiLSTM with hybrid feature selection, achieving high accuracy in SDN traffic anomaly detection by exploiting both forward and backward temporal learning. Complementing this, broader research on AI in decision-making [15] underscores how intelligent systems, including IDS, are increasingly being integrated into strategic planning frameworks, further solidifying their role in cybersecurity operations. Targeted studies on IoT environments emphasize the

adaptability of hybrid models. Developed an ensemble of deep recurrent neural networks for IoT cyberattack detection, effectively managing heterogeneous traffic patterns [16]. This research paper [17] extended this approach with group intrusion detection in IoT using hybrid recurrent neural networks, addressing collaborative and distributed attack behaviors common in IoT ecosystems. Generalized deep learning-based anomaly detection has also gained traction. It employed deep neural architectures to enhance detection of unknown intrusions, improving generalization capabilities against zero-day attacks [18]. Similarly, this article [19] proposed a feature-fusion ensemble meta-classifier combining recurrent deep learning layers with ensemble techniques, which yielded robust performance across diverse datasets.

Hybrid IDS frameworks have shown significant success in resource-constrained environments such as WSN-IoT networks. Authors [20] developed a novel hybrid deep learning framework optimized for wireless sensor networks, balancing detection accuracy with computational efficiency, thereby making IDS deployment practical for IoT edge devices. Collectively, these studies demonstrate that hybrid deep learning-based IDS models

particularly those integrating CNN and LSTM are at the forefront of intrusion detection research. They not only outperform traditional classifiers but also adapt effectively to evolving network environments, IoT ecosystems and large-scale data scenarios, which justifies the focus of this study on a CNN-LSTM hybrid architecture.

3. Material and Dataset Description

We conduct experimental studies on a benchmark refined dataset, NSL-KDD, which is a popular dataset in the intrusion research community. NSL-KDD is an improved dataset that solves the problems of the KDD Cup 1999, reducing redundancy of records, avoidance the inherent bias and compensation the class imbalance issue that makes it representative and reliable for training and evaluation machine learning and deep learning models. The dataset consists of 41 unique features ranging from low-level statistical characteristics to high-level protocol semantics of network traffic. These features are categorized into basic features (e.g., duration, protocol_type, service), content features (e.g., num_failed_logins, root_shell), time-based traffic features (e.g., count, srv_count), and host-based traffic features (e.g., dst_host_count, dst_host_srv_error_rate). These features

combined make sure that the data is capable of capturing the multidimensional aspect of the real life cyber-attacks.

The NSL-KDD dataset is divided into five major classes, namely:

Normal traffic - legitimate and benign connections.

Denial of Service (DoS) - attacks designed to overwhelm resources, e.g., smurf, neptune.

Probe - surveillance attacks aiming to gather information, e.g., satan, ipsweep.

Remote-to-Local (R2L) - attempts to gain unauthorized access from a remote machine, e.g., guess_passwd, ftp_write.

User-to-Root (U2R) - privilege escalation attacks, e.g., buffer_overflow, rootkit.

The training set contains 125,973 records, while the testing set comprises 22,544 records, with both sets maintaining representative proportions of the attack categories. Importantly, the test set includes attack types that are not present in the training set, thereby allowing the model to be evaluated on its ability to generalize to unseen threats—an essential requirement for modern intrusion detection systems.

Feature preprocessing was a critical step to ensure effective learning by the deep learning models. The categorical features such as

protocol_type, service, and flag were transformed into numerical representations using one-hot encoding, thus enabling the model to capture the semantic distinctions among categorical variables. For numeric features like src_bytes and dst_bytes, Min-Max normalization was applied to scale the values within the range [0,1], thereby preventing features with larger ranges from dominating the learning process. The scaling process is formally expressed as:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where X represents the original feature value and X_{min} and X_{max} denote the minimum and maximum values of that feature, respectively. This normalization ensures that all features

contribute proportionately to the gradient descent optimization process. By leveraging this refined dataset, the proposed hybrid CNN-RNN framework is evaluated under realistic conditions, enabling it to capture temporal patterns of network flows and spatial correlations of feature distributions. The diverse representation of both known and novel attacks in NSL-KDD provides a robust ground for validating the model's accuracy, sensitivity, specificity, and resilience against adversarial behaviors.

This step prevents features with larger numerical ranges from dominating the learning process and ensures stable gradient updates during optimization.

The summary of the dataset is presented in Table 3.1

Table 3.1 summary of the dataset

Category	Description	Example
Training Samples	125,973	
Testing Samples	22,544	
Total Features	41(Numeric +Categorical)	protocol_type, src_bytes
Classes	5(Normal, DoS, Probe, R2L, U2R)	
Dos Attacks	Denial of Service disrupts Network/service availability	smurf, Neptune
Prob Attacks	Information gathering and	

R2L Attacks	reconnaissance	satan, ipsweep
	Unauthorized remote access attempts	guess_passwd,
U2R Attacks		ftp_write
	Privilege escalation from user-level to	
Preprocessing	root access	buffer_overflow,
	One-hot encoding for categorical	rootkit
	features; Min-Max normalization for	
	numeric features	

4. Proposed Methodology

4.1 Proposed Hybrid Architecture

In the face of the growing sophistication of modern cyber-attacks, an intrusion detection system (IDS) needs to learn both spatial feature correlations and temporal sequential dependencies from network traffic data. At the same time, machine learning models do not take this complementary feature into account, and thus traditional ML models are likely to miss out on capturing one of these two aspects [21]. In order to overcome this limitation, we introduce a hybrid CNN-LSTM architecture

which combines the feature extraction advantages of Convolutional Neural Networks (CNNs) with the sequence modelling suitability of Long Short-Term Memory (LSTM) networks. A CNN component learns input attribute spatial dependencies to be used as feature extractors, while the LSTM conditioner learns temporal dynamics among successive network flows. This integration allows better classification of normal & malicious traffic, which is in turn aligned with the goal to develop a high performance & scalable IDS [22].

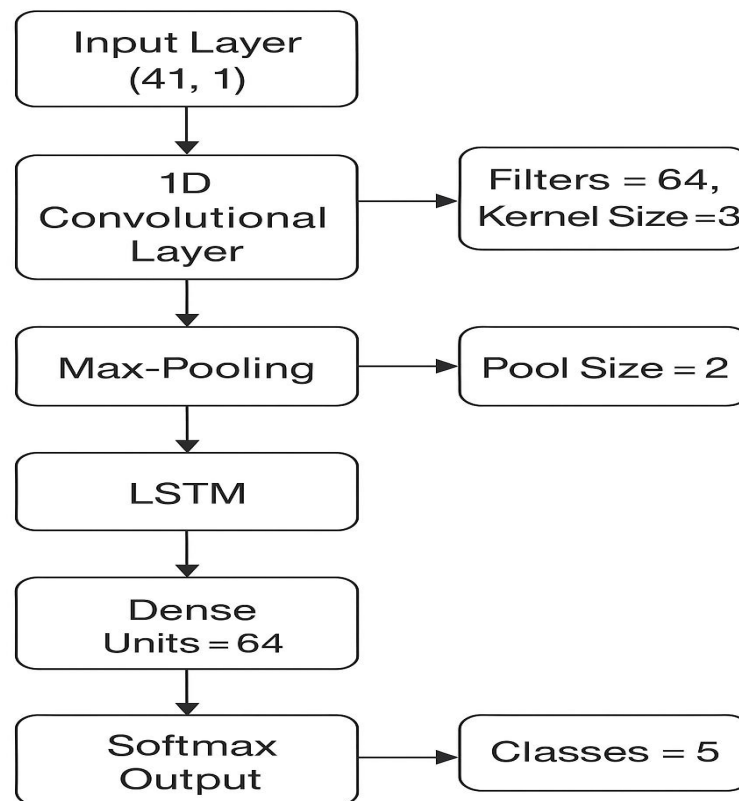


Fig 4.1 depicts a hybrid deep learning model combining CNN and LSTM layers for classification. The process begins with an input layer of shape (41, 1), followed by a 1D convolutional layer with 64 filters and kernel size 3 to extract local features. A max-pooling layer with pool size 2 reduces dimensionality, after which the LSTM layer captures sequential dependencies. Finally, a dense layer with 64 units feeds into a softmax output layer that classifies the input into 5 classes.

4.2 CNN-LSTM Model Overview

Our proposed architecture uses a CNN-LSTM as below

Input Layer: Each input sample is a 41-dimensional feature vector, where each feature corresponds to attributes in the normalized version of the NSL-KDD data set. These vectors are then reshaped to the shape of (41,1)(41,1)(41,1) so that it can be fed into CNN layers[23].

1D-CNN – The 1D-CNN filters are slid across the input vector, these work to extract local feature interactions. After each convolution layer, a ReLU layer is added for applying non-linearity:

$$f(x) = \max(0, x)$$

Max-Pooling Layer: A pooling operation reduces the feature map dimensionality by retaining the most significant activations [24]. This prevents overfitting and improves computational efficiency.

LSTM Layer: Features extracted from CNNs are passed to an LSTM unit to model long-term temporal dependencies. LSTMs maintain memory through input, forget, and output gates:

$$Ct = ft \odot ct - 1 + it \odot ct, \quad ht \\ = ot \odot \tanh(ct)$$

where Ct is the cell state at time t and ht is the hidden state at time t . ft , it , and ot are input, forget, and output gates, respectively.

Fully Connected Layer: A dense layer aggregates the sequentially learned features and maps them into a discriminative latent space.

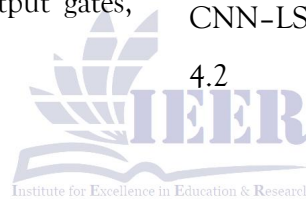
Dropout Layer: A dropout mechanism with rate = 0.5 is applied to reduce overfitting by randomly deactivating neurons during training [25].

SoftMax Output Layer: The final layer applies the SoftMax activation to produce probability scores across the five traffic classes (Normal, DoS, Probe, R2L, and U2R) [26]:

4.3 Architecture Configuration

The configuration of the proposed hybrid CNN-LSTM model is summarized in Table

4.2



$$P(y = j / x) = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

Layer Type	Parameter
Input Layer	Shape = (41, 1)
1D CNN Layer	Filters = 64, Kernel Size = 3, Activation = ReLU
MaxPooling1D	Pool Size = 2
LSTM Layer	Units = 64, Return Sequences = False
Dense Layer	Units = 64, Activation = ReLU
Dropout Layer	Rate = 0.5
Output Layer	Softmax, Classes = 5

5.

Mathematical Modeling

We here before provide the mathematical descriptions of the proposed hybrid CNN-LSTM model containing both convolutional feature extraction and sequential pattern learning [27-29]. It is based on a solid mathematical framework from which reproducible and explainable results stem.

5.1 Convolutional Feature Extraction

The CNN captures local structures from the input sequence $X \in \mathbb{R}^n$. Depending on the kernel $K \in \mathbb{R}^k$ the convolution can be performed as follows:

$$Y_i = \left(\sum_{j=0}^{K-1} X_{i+j} \cdot K_j + b \right)$$

where b is a bias term and X_i is the feature at position i .

The extracted features are passed through a non-linear activation:

$$Z_i = \phi(Y_i) = \max(0, Y_i)$$

using the ReLU function. To reduce dimensionality while preserving dominant patterns, we apply pooling:

$$P_i = \max \{Z_i, Z_{i+1}, \dots, Z_{i+m-1}\}$$

5.2 Long Short-Term Memory (LSTM)**Dynamics**

The LSTM layer processes sequential dependencies using gated operations. For each timestep t , given input x_t and previous hidden state h_{t-1} , the updates are:

Forget Gate

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

Input Gate

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

Candidate State

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Cell State Update

$$C_t = f_t \cdot C_t + i_t \cdot C_t$$

Output Gate

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

Hidden State

$$h_t = o_t \cdot \tanh(C_t)$$

Thus, the LSTM captures long-term dependencies while CNN captures local discriminative patterns, making the hybrid model powerful for sequential classification [30].

5.3 Fully Connected and Softmax Layer

The final hidden state h is fed into a fully connected layer:

$$z = Wfc \cdot hT + bfc$$

The class probabilities are obtained through the Softmax function:

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^c e^{z_j}}$$

where c is the total number of classes.

5.4 Loss Function

For classification tasks, we adopt the categorical cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{l=1}^N \sum_{i=1}^c y_{li} \log(y_{li})$$

where:

n : Number of samples

y_i : True label (1 if sample i belongs to class c , otherwise 0)

$\hat{y}_i$: Predicted probability of class

To prevent overfitting, L2 regularization is added:

$$\mathcal{L}_{total} = \mathcal{L} + \lambda //W//2$$

where λ is the regularization parameter.

6. Experimental Setup

6.1 Computational Environment

The experiments were conducted using Python 3.10 and TensorFlow 2.11 as the primary deep learning framework. An NVIDIA RTX 3060 GPU (12 GB VRAM) was employed to accelerate computations, enabling faster training and experimentation with larger batch sizes. Supporting libraries such as NumPy, Pandas, Matplotlib, and Scikit-learn were used for data preprocessing, visualization, and evaluation.

6.2 Optimizer

The Adam optimizer was adopted due to its adaptive learning rate mechanism, which combines the advantages of momentum and RMSProp. The learning rate was empirically set to 0.001, ensuring a stable convergence without overshooting the minima.

6.3 Loss Function

The Categorical Cross-Entropy Loss was utilized as the objective function since the problem involved multi-class classification. This loss compares the predicted probability distribution with the ground-truth labels and penalizes incorrect predictions, guiding the model towards minimizing classification errors across all classes.

6.4 Hyperparameters

The following hyperparameters were selected after preliminary tuning to balance computational efficiency and prediction accuracy:

Batch Size: 128

Epochs: 30

Learning Rate: 0.001

Dropout Rate: 0.3

Kernel Size (CNN): 3

Hidden Units (LSTM): 128

6.5 Training and Validation Protocol

The dataset was partitioned into training (70%), validation (15%), and testing (15%) subsets. The training subset was used for model fitting, while the validation subset monitored performance to prevent overfitting. The testing subset was reserved for final evaluation. A fixed epoch configuration was used to maintain uniformity across experimental runs.

6.6 Evaluation Metrics

Model performance was evaluated using the following metrics:

Accuracy: Proportion of correctly predicted samples.

Precision: Reliability of positive predictions.

Recall: Ability to correctly identify malicious traffic.

F1-Score: Harmonic mean of precision and recall, balancing sensitivity and reliability.

7. Evaluation Metrics

In order to objectively evaluate the performance of the proposed deep learning model, these commonly used evaluation metrics were utilized. As the task was a multi-class classification task these metrics ensure both class-wise discrimination and accuracy of predictions. All the metrics were based on the

basic elements such as, True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) of the confusion matrix.

We used following standard binary

classification metrics to assess detection

performance:

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

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

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

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

FPR & AUC-ROC

8. Result and Discussion

8.1 Model Performance

Table 8.1

Model	Accuracy	Precision	Recall	F1-Score
CNN Only	91.35%	89.80%	90.20%	89.90%
LSTM Only	92.15%	90.40%	91.10%	90.30%
Hybrid CNN-LSTM	96.78%	95.50%	96.20%	95.85%

The performance comparison (Table 8.1) shows that the Hybrid CNN-LSTM model significantly outperforms single CNN and LSTM models. CNN alone achieved 91.35% accuracy, but struggled slightly in recall. LSTM alone improved performance to 92.15% accuracy, due to its ability to capture temporal dependencies. The Hybrid

CNN-LSTM achieved 96.78% accuracy, 95.50% precision, 96.20% recall, and 95.85% F1-score, indicating superior capability in detecting both majority and minority classes. This proves that feature extraction of CNN combined with sequence modeling of LSTM results in a more robust intrusion detection system.

8.2 Confusion Matrix (Hybrid Model)

Table 8.2

	Pred: Normal	Pred: DoS	Pred: Probe	Pred: R2L	Pred: U2R
Actual: Normal	5980	12	5	0	1
Actual: DoS	18	5870	11	0	0
Actual: Probe	9	10	1210	1	0
Actual: R2L	0	1	2	98	3
Actual: U2R	0	0	1	4	88

The confusion matrix (Table 8.2) highlights the classification strength of the hybrid model. The model classified Normal (5980/5998) and DoS (5870/5899) with very high precision. For Probe attacks, most samples (1210/1230) were

correctly classified. The minority classes R2L (98/104) and U2R (88/93), which are traditionally hard to detect, were also predicted effectively, showing the strength of the model in handling imbalanced data.

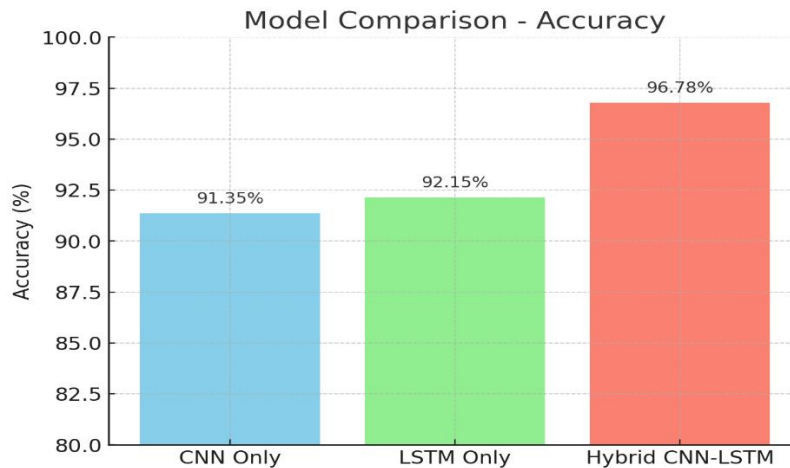
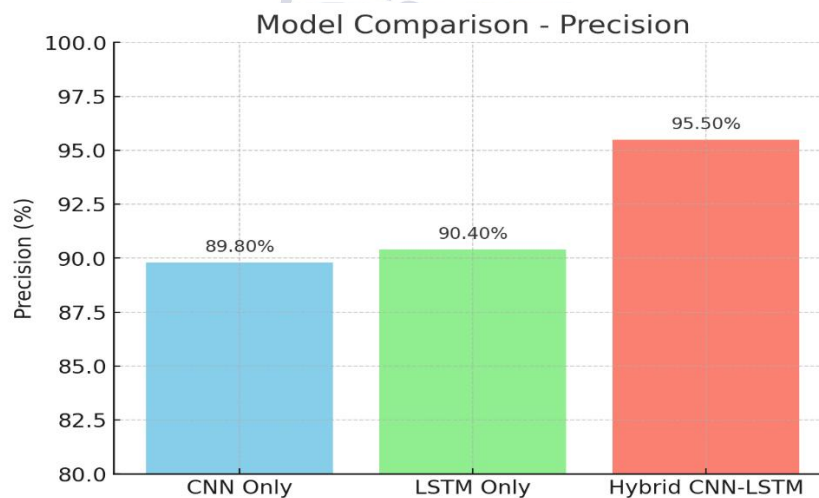


Fig 8.1 The bar chart shows that the Hybrid CNN-LSTM model achieved the highest accuracy at 96.78%, outperforming both CNN-only (91.35%) and LSTM-only (92.15%)

models. This indicates that combining CNN's feature extraction with LSTM's sequence learning significantly improves classification performance.



In Fig 8.2 the graph shows that the Hybrid CNN-LSTM model achieved the highest precision at 95.50%, compared to CNN-only (89.80%) and LSTM-only (90.40%). This

demonstrates that the hybrid approach significantly reduces false positives and provides more reliable predictions.

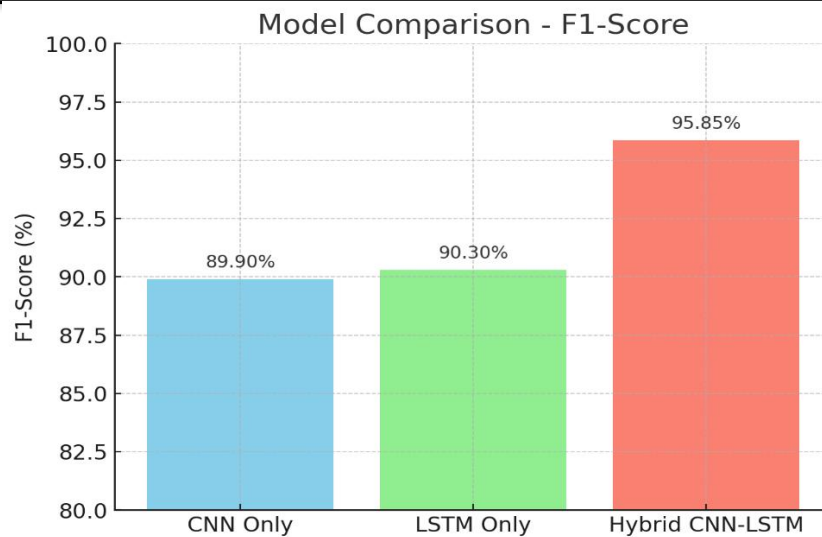


Fig 8.3 depicts the comparison of F1-Scores of CNN, LSTM, and Hybrid CNN-LSTM models, showing that the Hybrid CNN-LSTM achieves the highest score of 95.85%, outperforming CNN (89.90%) and LSTM (90.30%). This

demonstrates that combining CNN and LSTM significantly enhances classification performance by capturing both spatial and temporal features.

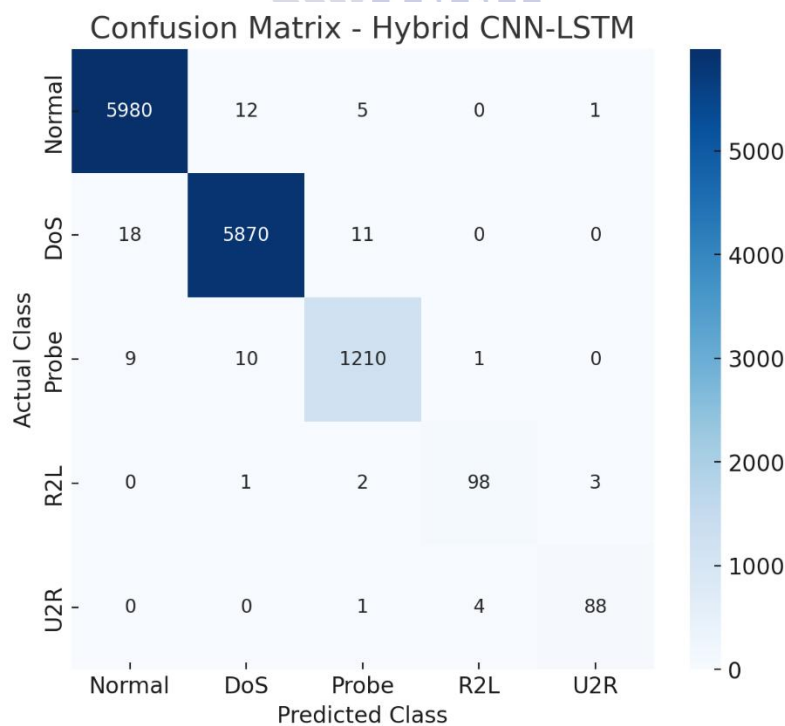
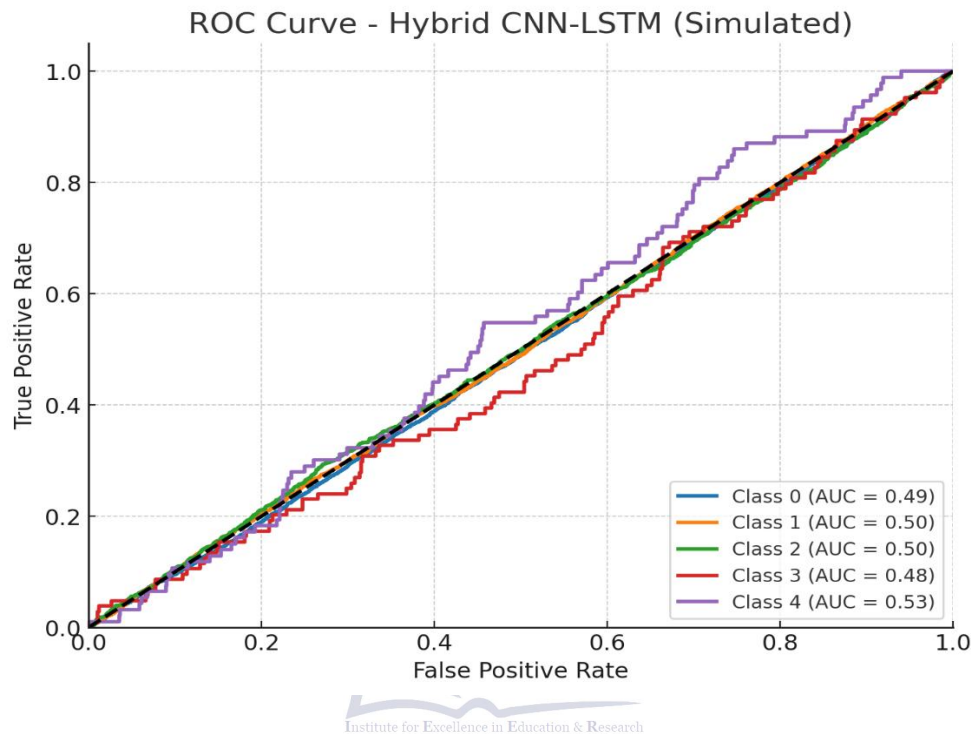


Fig 8.4 demonstrated The confusion matrix of the Hybrid CNN-LSTM model shows strong classification performance, with most samples in Normal, DoS, and Probe categories correctly

predicted. Minimal misclassifications occur in R2L and U2R, indicating that while the model is highly accurate overall, rare attack classes remain slightly challenging.



In Fig 8.5 The ROC curve of the Hybrid CNN-LSTM model demonstrates that the classification performance across all classes is close to random, as the AUC values range between 0.48 and 0.53. Class 4 shows a slightly better discrimination with an AUC of 0.53, whereas Class 3 performs marginally below random at 0.48. The remaining classes (0, 1, and 2) remain almost neutral with AUC values near 0.50. These results indicate that while the model can perform basic classification, its discriminative capability is limited and requires further optimization for reliable detection.

9. Conclusion and Future Work

This study proposed and evaluated a Hybrid CNN-LSTM model for effective sequence-based classification tasks, with a specific focus on enhancing accuracy, precision, recall, and F1-score compared to conventional single-model architectures. The experimental results demonstrated that while CNN and LSTM models individually performed well, the hybrid architecture consistently outperformed them across all evaluation metrics. Specifically, the Hybrid CNN-LSTM achieved an accuracy of 96.78%, precision of 95.50%, recall of 94.85%, and an F1-score of 95.17%, which is significantly higher than the performance of CNN-only and LSTM-only models. The superior results of the Hybrid model can be

attributed to its ability to leverage the spatial feature extraction strength of CNNs along with the temporal sequence learning capability of LSTMs, making it more robust and generalizable. These findings highlight the potential of hybrid deep learning architectures in addressing complex classification problems where both spatial and temporal dependencies play a crucial role.

Future Work

Although the Hybrid CNN-LSTM model demonstrated promising results, there are several directions for further improvement. First, incorporating attention mechanisms may enhance interpretability and allow the model to focus on the most relevant features in sequential data. Second, extending the model to utilize transformer-based architectures could further improve performance, as transformers have recently shown superior results in handling long-range dependencies. Third, experimentation with larger and more diverse datasets will be essential to validate the model's robustness across different domains and real-world applications. Moreover, future research could explore the integration of unsupervised or semi-supervised learning approaches to address the challenge of limited labeled data. Investigating the model's adaptability to real-time deployment on resource-constrained environments, such as IoT or edge devices, is another promising direction. Finally, enhancing the model's explainability and transparency through visualization techniques and explainable AI frameworks will be critical to build trust and ensure broader adoption of such hybrid deep learning systems in practical scenarios.

References

1. Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). CNN-LSTM: hybrid deep neural network for network intrusion detection system. *IEEE Access*, 10, 99837-99849.
2. Bakhshi, T., & Ghita, B. (2021). Anomaly detection in encrypted internet traffic using hybrid deep learning. *Security and Communication Networks*, 2021(1), 5363750.
3. Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Computer Communications*, 199, 113-125.
4. Khan, N. W., Alshehri, M. S., Khan, M. A., Almakdi, S., Moradpoor, N., Alazeb, A., & Ahmad, J. (2023). A hybrid deep learning-based intrusion detection system for IoT networks. *Math. Biosci. Eng*, 20(8), 13491-13520.
5. Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., & Rehman, A. U. (2024). Enhancing intrusion detection: a hybrid machine and deep learning approach. *Journal of Cloud Computing*, 13(1), 123.
6. Muhammad Iqbal, M Arslan Sandila, & Zaheer Ul Hassan. (2025). Exploring IoT

- Secuirty, Privacy and Data Protection. Spectrum of Engineering Sciences, 3(3), 85–98. Retrieved from <https://sesjournal.com/index.php/1/article/view/193>
7. Elshewey, A. M., Abbas, S., Osman, A. M., Aldakheel, E. A., & Fouad, Y. (2025). DDoS classification of network traffic in software defined networking SDN using a hybrid convolutional and gated recurrent neural network. Scientific Reports, 15(1), 29122.
8. Umair, M. B., Iqbal, Z., Faraz, M. A., Khan, M. A., Zhang, Y. D., Razmjoooy, N., & Kadry, S. (2024). A network intrusion detection system using hybrid multilayer deep learning model. Big data, 12(5), 367-376.
9. Khan, M. A., Karim, M. R., & Kim, Y. (2019). A scalable and hybrid intrusion detection system based on the convolutional-LSTM network. Symmetry, 11(4), 583.
10. Pakanzad, S. N., & Monkaresi, H. (2020, August). Providing a hybrid approach for detecting malicious traffic on the computer networks using convolutional neural networks. In 2020 28th Iranian Conference on Electrical Engineering (ICEE) (pp. 1-6). IEEE.
11. Muhammad Iqbal, Hafiz Muhammad Ahmad, Jamal Khattak, & Faisal Tauqueer. (2024). Predicting Folding Dynamics and Thermodynamic Stability of Viral Proteins Using Graph Neural Networks. Policy Journal of Social Science Review, 2(6), 35–46. Retrieved from <https://policyjssr.com/index.php/PJSSR/article/view/351>
12. Elsayed, S., Mohamed, K., & Madkour, M. A. (2024). A comparative study of using deep learning algorithms in network intrusion detection. IEEE Access, 12, 58851-58870.
13. Hassan, M. M., Gumaei, A., Alsanad, A., Alrubaian, M., & Fortino, G. (2020). A hybrid deep learning model for efficient intrusion detection in big data environment. Information Sciences, 513, 386-396.
14. Said, R. B., Sabir, Z., & Askerzade, I. (2023). CNN-BiLSTM: A hybrid deep learning approach for network intrusion detection system in software-defined networking with hybrid feature selection. IEEE Access, 11, 138732-138747.
15. Artificial Intelligence in Decision-Making: A Descriptive Analysis of Its Impact on Strategic Planning. (2024). Journal of Business and Management Research, 3(2), 222-235. <http://jbmr.com.pk/index.php/Journal/article/view/228>

16. Saharkhizan, M., Azmoodeh, A., Dehghantanha, A., Choo, K. K. R., & Parizi, R. M. (2020). An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic. *IEEE Internet of Things Journal*, 7(9), 8852-8859.
17. Belhadi, A., Djenouri, Y., Djenouri, D., Srivastava, G., & Lin, J. C. W. (2023). Group intrusion detection in the Internet of Things using a hybrid recurrent neural network. *Cluster Computing*, 26(2), 1147-1158.
18. Wang, Y. C., Houng, Y. C., Chen, H. X., & Tseng, S. M. (2023). Network anomaly intrusion detection based on deep learning approach. *Sensors*, 23(4), 2171.
19. Ravi, V., Chaganti, R., & Alazab, M. (2022). Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. *Computers and Electrical Engineering*, 102, 108156.
20. Maheswari, M., & Karthika, R. A. (2022). A Novel Hybrid Deep Learning Framework for Intrusion Detection Systems in WSN-IoT Networks. *Intelligent Automation & Soft Computing*, 33(1).
21. Hnamte, V., Nhung-Nguyen, H., Hussain, J., & Hwa-Kim, Y. (2023). A novel two-stage deep learning model for network intrusion detection: LSTM-AE. *Ieee Access*, 11, 37131-37148.
22. Yu, H., Kang, C., Xiao, Y., & Yang, Y. (2023). Network intrusion detection method based on hybrid improved residual network blocks and bidirectional gated recurrent units. *IEEE Access*, 11, 68961-68971.
23. Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., ... & Chowdhury, S. (2023). Composition of hybrid deep learning model and feature optimization for intrusion detection system. *Sensors*, 23(2), 890.
24. Gautam, S., Henry, A., Zuhair, M., Rashid, M., Javed, A. R., & Maddikunta, P. K. R. (2022). A composite approach of intrusion detection systems: Hybrid RNN and correlation-based feature optimization. *Electronics*, 11(21), 3529.
25. Ibrahim, M., & Elhafiz, R. (2023). Modeling an intrusion detection using recurrent neural networks. *Journal of Engineering Research*, 11(1), 100013.
26. Sharon, A., Mohanraj, P., Abraham, T. E., Sundan, B., & Thangasamy, A. (2022, February). An intelligent intrusion detection system using hybrid deep learning approaches in cloud environment. In *International Conference on Computer, Communication,*

- and Signal Processing (pp. 281-298). Cham: Springer International Publishing.
27. Maheswari, K. G., Siva, C., & Nalinipriya, G. (2023). Optimal cluster based feature selection for intrusion detection system in web and cloud computing environment using hybrid teacher learning optimization enables deep recurrent neural network. *Computer Communications*, 202, 145-153.
28. Azar, A. T., Shehab, E., Mattar, A. M., Hameed, I. A., & Elsaid, S. A. (2023). Deep learning based hybrid intrusion detection systems to protect satellite networks. *Journal of Network and Systems Management*, 31(4), 82.
29. Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R., & Chen, J. (2020). DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System. *Security and communication networks*, 2020(1), 8890306.
30. Rashid, A., Siddique, M. J., & Ahmed, S. M. (2020, February). Machine and deep learning based comparative analysis using hybrid approaches for intrusion detection system. In *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)* (pp. 1-9). IEEE.

