National Research Journal of Information Technology & Information Science
Volume No: 12, Issue No: 2, Year: 2025 (July- December)
PP: 257-271
ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

# INTELLIGENT NETWORK TRAFFIC CLASSIFICATION: DEEP LEARNING INTEGRATION AND COMPARATIVE INSIGHTS

**Gurpreet Kaur**

School of Engineering and Technology, CT University, Ludhiana, Punjab, India

**Arvind Kumar**

School of Engineering and Technology, CT University, Ludhiana, Punjab, India

**Kamal Malik**

Computer Science and Engineering, Maharishi Markandeshwar (Deemed to be University), Mullana- Ambala, Haryana, India

## ABSTRACT

Due to rapid advancements in network technologies, the digital landscape has become quite complex. Accurate prediction of future traffic patterns have become crucial and critical for optimal resource allocation and effective network management. So, in response to the challenge, this study introduces a comprehensive deep learning framework that is designed to enhance the accuracy of network traffic classification. Through the integration of CNN-based spatial feature extraction and LSTM-driven temporal modeling, the hybrid design enhances classification accuracy, compared to the conventional methods. The proposed model is applied to WSN-DS dataset that achieved an impressive classification accuracy of 98%, verified using standard performance evaluation metrics. A comprehensive comparative analysis against state-of-the-art techniques further demonstrated its enhanced effectiveness across multiple performance indicators. These outcomes highlight the framework's robustness, scalability, and practical applicability, establishing it as a promising solution for real-time network monitoring and intelligent traffic analysis in increasingly complex network environments.

**KEYWORDS:**

Deep learning, network traffic prediction, convolutional neural networks, long-short term memory, wsn-ds dataset

## 1. INTRODUCTION

The adoption of network based applications in various fields has been made possible due to the rapid development of internet. As a result, the network traffic is becoming more complex and heterogeneous and its behavior is not only important to be known but also challenging to model [1]. Simultaneously, the emergence of cyber-attacks and specific dangers of wireless communication technologies has caused the increase of anxiety in the circles of private organizations and governmental institutions across the globe [2]. Therefore, network traffic is of great informational value. Various analytical tools and methods have been created to easily process the huge amount of network data. Network traffic classification is one of them because it is critical in the aspect of knowing the prevailing conditions of the network and the possibility of early identification of anomalous practices in the network before the organizations can take the initiative to protect their systems. Network traffic classification can be defined as the process of classifying traffic-generating activities in a network. It is an essential part of a proper network management and security maintenance [3]. Moreover, traffic classification helps in the identification of various kinds of network applications so that network resources can be allocated with ease and with optimum efficiency [4]. In order to do so, machine learning algorithms are

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

typically utilized to handle and classify network data packets by eliciting the most discriminable features [5]. Some of the widely used popular classifiers in network traffic classification are the Decision Trees (DT), Random Forests (RF), Bayesian models, Support Vector Machines (SVM), and the K-Nearest Neighbors (KNN). The en-bloc learning techniques also improve the accuracy of the prediction by integrating several base classifiers but dependencies among them should be seen as they also affect the overall performance greatly [6-7]. Although these algorithms can be very efficient compared to standard methods, they usually need human-crafted feature extraction and access to personal traffic information which makes them more restricted in extrapolation and cross-domain use. Also, classification methods based on machine learning often require a large amount of storage and CPU computation to operate and cannot be used on devices with limited resources. In addition, the majority of models are trained with stagnant or old data, which restricts their application in dynamic settings (e.g. encrypted protocols, IoT networks, new cloud services). Other key gap is despite the fact that ensemble methods enhance accuracy, they introduce dependencies between the base classifiers, complicating the algorithms and even decreasing the performance of the algorithms because of correlated errors. There is also a lack of explainability and interpretability in different traffic classification frameworks. The limitations emphasize the importance of more efficient and a less resource-intensive way of classification [8].

### 1.1. Research Hypothesis

Based on these gaps, this study hypothesizes that:

**Hypothesis 1**

$H_0(Null\ Hypothesis)$: CNN-based models are not much more accurate at opting to classify the data compared to conventional machine learning algorithms (e.g., SVM, RF, KNN).

$H_1(Alternative\ Hypothesis)$: CNN-based models exhibit a much improved classification accuracy compared to the conventional machine learning.

**Hypothesis 2**

$H_0(Null\ Hypothesis)$: LSTM-based models fail to increase classification accuracy after trying to capture temporal dependencies of network traffic sequences in the network.

$H_1(Alternative\ Hypothesis)$: LSTM-based models enhance the accuracy of classifications since they are time-dependent network traffic sequences.

**Hypothesis 3**

$H_0(Null\ Hypothesis)$: CNN-LSTM hybrid models fail to perform better in terms of classification accuracy in comparison to standalone CNN or LSTM models.

$H_1(Alternative\ Hypothesis)$: CNN-LSTM hybrid models perform better classifications as compared to CNN or LSTM models only.

### 1.2. Research objectives

This study is guided by the following objectives:

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

1. To characterize the behavior of deep learning models on detecting and characterizing network traffic and assess their capability to automatically tease out useful features without engineering.
2. To develop a powerful and hybrid CNN-LSTM network that synthesizes the spatial feature, and the temporal sequence learning to identify intrusion in the network and classify the traffic.
3. To compare the model with the state-of-art models, with generalized measures such as accuracy, precision, recall and F1-score.

### 1.3. Contributions of the study

Deep Learning (DL) is one of the high-tech approaches. It is biased to adopt the relevant features in the process of training. Such automation saves much of human efforts and improves the performance of the model. Although the training data is complicated and requires optimization of the hyper-parameters, the results obtained by DL are good with regard to its accuracy. Moreover, deep learning has been found to be more accurate than the classical ML algorithms, including KNN, RF, and SVMs, and can be used to integrate a combination of multiple algorithms into a single standard model, including network intrusion detection (NID), critical link analysis, and traffic classification. These strengths have made it extensively used in network related applications especially in traffic classification. The most effective strategies include Convolutional Neural Networks (CNNs), which are particularly effective to use in network traffic classification assignments since it is effective to capture local spatial features in sequential data and extracts features automatically [9]. CNNs can learn the temporal relations of packets indirect through the chain of analysis, which is aided by LSTM. Such a hybrid fusion of CNN and LSTM also enhance the accuracy of classification. So, this study makes the following contributions:

- It suggests a deep learning, feature-based system of network traffic classification, which removes the aspect of manual feature engineering. The proposed architecture acquires the discriminative features using the raw traffic data, and it is thus able to strongly identify the intrusion and malicious activity.
- It constructs a hybrid CNN-LSTM network combining 1D-CNN layers to extract local spatial features and LSTMs layers to learn long-term temporal dependencies between packets sequences.
- It performs an extensive performance analysis of the suggested CNN-LSTM model against the state-of-the-art traditional ML and deep learning methods.

### 1.4. Organization of the paper

The subsequent sections of this paper are structured as follows. Section 2 covers the main findings and contributions of the researchers. Section 3 covers the research methodology. Section 4 discusses the results and the proof of the approach. Section 5 compares and contrasts the proof with the cutting-edge techniques, and the last Section gives the concluding thoughts and recommendations of the paper.

## 2. MAIN FINDINGS AND RESEARCHERS CONTRIBUTIONS

Many Researchers investigated a wide range of ML strategies along with classification methods for attack detection. Few of them from the year 2018 to 2025 in chronological order are discussed below in the following **Table 1.**

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

**Table 1. Main Findings and its Contributions**

| Ref. | Dataset | Feature Selection/Classification Technique | Attack Detected | Detection Method | Accuracy | Remarks |
|---|---|---|---|---|---|---|
| 2025 [10] | Simulated dataset based on IoTID20 | Random Forest | Scan, DOS, Mirai and MITM | Genetic Algorithm | 96.5% | GA-inspired feature selection method is a simplified method that doesn't fully utilize the search capabilities of advanced evolutionary algorithms |
| 2024 [11] | Masked input dataset | Residual Network (ResNet)/ XAI (Explainable AI) | black box adversarial attack | Genetic Algorithm | 82.97% | Its efficacy may be limited by the diversity of adversarial attacks it has been trained on, and its generalization capability across different models and datasets |
| 2023 [12] | UNSW-NB15, NSL-KDD InSDN dataset | RF + RFE/ Random Forest | R2L, DoS, U2R, Probe, Web Attack, Botnet and many more | CNN + BiLSTM | 93.5, 95.9, and 97.7% respectively | Promising approach for attack detection that can effectively detect most types of attacks with high accuracy and precision |
| 2022 [13] | NSL-KDD | Correlation-based feature selection | DOS, R2L, U2R and Probing | RF, J48, SVM and NB | 87 to 97 % for different classifiers | Results show that Random Forest performed well compared to the other models in predicting the malicious packets |
| 2021 [14] | NSL-KDD and CICIDS 2017 | Entropy decision feature selection | High detection rate | Decision Tree with enhanced data quality | 99 and 98% respectively | Achieved excellent accuracy but can be integrate with deep learning techniques to further improve detection rate |
| 2020 [15] | Software Defined Network (SDN) testbed | Tcpreplay tool | Seven popular applications | MLP, SAE, CNN | 93% approx. | Due to the hardware limitation, possible statistics packets were dropped, when the testing dataset injected into the OVS switch according to the same |

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | processing time and sequence |
| 2019 [16] | NSL-KDD | Stacked Autoencoder (SAE) + CNN | DOS, R2L, U2R and Probing | CNNs + LSTMs | 76 to 83% | Good performance but can be improved by increasing no. of hidden layers or neurons |
| 2018 [17] | KDD'99, NSL-KDD | Automatic feature extraction | Denial of Service attacks | Recurrent Neural Network | 99% approx. | Although superior accuracy but outdated datasets are used |

## 3. RESEARCH METHODOLOGY

This section discusses the challenge of network traffic classification that is divided into number of steps. Previous studies have highlighted the limited accuracy of the models in classifying network traffic. A new model has been put forth to overcome this drawback. It leverages integrated approach which combines CNNs and LSTM methods for classification of network traffic. Following subsections explains proper procedure followed for implementing the proposed model.

### 3.1. Data Acquisition

The WSN-DS is a dataset that is hosted in Kaggle and is commonly used to analyze and identify regular network attacks, which include Grayhole, Blackhole, and Flooding. Every record within the dataset has several features based on network traffic such as the properties of packets such as size, protocol and the number of bytes being sent and received.

### 3.2. Data Preprocessing

To prevent the learning process from becoming dominated by features with large values, the numerical features are scaled to a common range 0-1. The Z-score normalization procedure is then used to transform these features. The standard formula is: $X\_scaled = (X - \mu) / \sigma$.

### 3.3. Principal Component Analysis

The Principal Component Analysis (PCA) is a technique for dimensionality reduction that converts the original features into a new set of uncorrelated principal components. This type of feature classification can be formally represented by the given dataset's representation. In this method, the first principal component of the data captures the biggest variance, the second captures the next largest variance while remaining orthogonal to the first, and subsequent components follow the same pattern. It can be formally represented in the given dataset as: $X = \{x_1, x_2, x_3, \ldots x_n\}$ where, $x_i \in R^d$ (1)

The representation of the data samples in a given dataset is composed of vectors of features. The dimension of the real space where these features are located is called $R^d$ which is the d-dimensional real space where each data vector has d features. The projection onto lower dimensional space is defined as: $z = W_k^T x$, $z \in R^r$, $r<d$ (2)

where, $W_k^T$ is the transpose of projection matrix. Now, variance of projected data is maximized as: $\max (z) = w_1^T X^T X w_1$ (3)

subject to the orthogonality constraint $w_i^T w_j = 0$ for $i \neq j$.

The computation of the eigenvalue of the $X^T X$ matrix is carried out under the constraints of the

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

orthogonality constraint. The first component of the matrix has the highest eigenvalue, and the others are similar. It is evident that the PCA algorithm generates a reduced set of features while retaining the majority of the data's variance.

### 3.4. Integrated Development

In this stage, a deep learning model is developed by incorporating CNNs with an LSTM network. CNNs is utilized to extract high and low-level spatial features from data. By leveraging convolutional operations, the model effectively identifies local dependencies and spatial correlations within the reduced feature set obtained after preprocessing and dimensionality reduction. If $l^{th}$ layer is a convolution layer, then the convolution output for $i^{th}$ position and $j^{th}$ filter can be written as:

$$x_i^{(l,j)} = f\left(\sum_{m=1}^{M(l-1)} \sum_{n=0}^{k-1} w_n^{(l,j,m)} x_{i_n}^{(l-1,m)} + b^{(l,j)}\right) \quad (4)$$

Output of neuron j in layer l for input index i. Relu is the activation function denoted by f. No. of neurons or feature maps in the previous layer l-1 is denoted by M. 'k' is the size of kernel. 'w' denotes the weight at position n of filter j corresponding to channel m. 'b' is the bias term that has been taken for filter j. Here, max pool method is used for pooling layer in which pooled output at position i in layer l is equal to maximum value in the pooling window keeping in view 'k' and 's' which are kernel size and step size respectively. This is written in the equation below:

$$p_i^{(l)} = max_{0 \le n < k} x_{i.s+n}^{(l-1)} \quad (5)$$

For 'k' number of classes and input vector of scores ($z_1, z_2, z_3 \ldots$). The output class's probability is calculated using the Softmax function which is written in equation (6):

$$\sigma(z)_i = \frac{e^z}{\sum_{j=1}^{k} e^z} \quad (6)$$

Cross entropy loss in the form of predicted probability for class i, is written in equation (7):

$$L(y, \sigma(z)) = -\sum_{i=1}^{k} y \, (\log(\sigma(\boldsymbol{z})_i) \quad (7)$$

After the spatial features are extracted, they are forwarded to the LSTM layer, which is tailored to capture long-term sequential and temporal dependencies. This step is essential for accurately modeling the time-series characteristics of network traffic in wireless sensor networks (WSNs). The overall process is illustrated in **Figure 1**

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

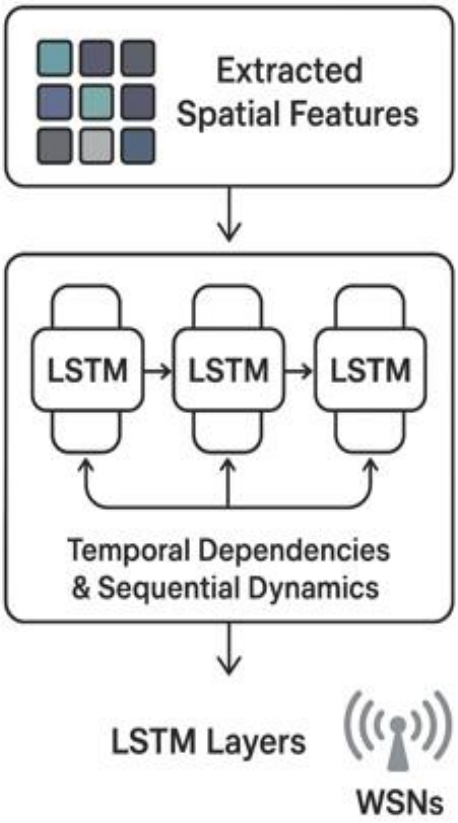


**Figure 1 LSTM process**

At each time step t: $h_t = \text{LSTM}(x_t, h_{t-1})$

The traffic or input data is referred to as xt, and the LSTM takes note of the hidden state ($h_{t-1}$). The network preserves past information and integrates it with the current input. In an LSTM block, the current input and hidden state are processed together, while the input, output, and forget gates govern the flow of information by controlling what is retained, updated, or discarded from earlier states. They also determine the appropriate response to the new input. So, each LSTM layer processes sequential patterns over time, learning temporal dependencies to detect anomalies, predict traffic and classify patterns effectively.

### 3.5. Optimizer selection

To ensure optimal performance of the proposed model, Adam optimizer has been used. The Adam (Adaptive Moment Estimation) optimizer is widely used for deep learning tasks because it combines the benefits of AdaGrad (adaptive learning rates per parameter) and RMSProp(momentum on squared gradients). The major advantage of using Adam in this hybrid approach ia that it adjusts learning rates separately for each layer's weights, thereby reducing its instability and training time. The optimizer was applied using its default learning rate settings to maintain consistency, and the model was trained with the configuration setting (epochs=20, validation steps=100).

### 3.6. Hyperparameter tuning

Hyperparameters used for hybrid CNN + LSTM model are depicted below in **Table 2.**

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

**Table 2. Hyperparameter for hybrid model**

| Parameter | Value/Description |
|---|---|
| Optimizer | Adam |
| Learning rate | 0.001 |
| Loss function | Cross entropy |
| Validation steps | 100 |
| Validation split | 20% of training data |
| Convolution type | 1D CNN |
| No. of convolution layers | 2 |
| Filters | 64 |
| Kernel size | 3 |
| Stride | 1 |
| Activation | ReLU |
| Pooling | MaxPool1D(pool size=2) |
| Units | 128 |
| Dense neurons | 64 |
| Activation hidden | ReLU |
| Output activation | Sigmoid |
| Dropout | 0.3 |

### 3.7. Performance Evaluation

After selecting the best-performing optimizer and identifying the optimal hyperparameters (learning rate and dropout rate), the proposed model was trained with these configurations to achieve optimal performance. The model was trained for 20 epochs with 100 steps per epoch for the training set and 100 validation steps for the validation set. Early stopping was implemented to prevent overfitting. To comprehensively assess the performance of the proposed model a diverse set of evaluation metrics was employed on the test dataset. These metrics are listed in **Table 3**.

**Table 3. Metrics for performance evaluation**

| Metric | Equation/ Description |
|---|---|
| Accuracy | $\frac{\text{Number of points correctly classified}}{\text{Total Number of points}} * 100$ |
| Precision | $\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$ |
| Recall | $\frac{True\ Positive}{True\ Positive + False\ Negative}$ |
| F1- score | $2 * \frac{Precision\ X\ Recall}{Precision + Recall}$ |

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

In addition to these metrics, the Receiver Operating Characteristic- Area Under Curve (ROC AUC) score was calculated that evaluates the model's ability to distinguish between classes. A deeper analysis of the model's performance was conducted by examining the confusion matrix.

## 4. RESULTS AND DISCUSSION

The proposed model uses the CNN-LSTM model to train on a pre-processed and PCA-reduced dataset. It is then evaluated to determine its performance in identifying network traffic as malicious or normal. The results are presented in the following sections.

### 4.1 Implementation Results

The model is implemented on a system with a Core i7 processor. It uses a main frequency of around 2.80 GHz. and equipped with 32 GB of RAM**.** The operating system environment is Windows 10, and it is carried out using Python 3.7 and TensorFlow 2.2.0 as the primary experimental tools. The class distribution for different classes is shown in **Figure 2**

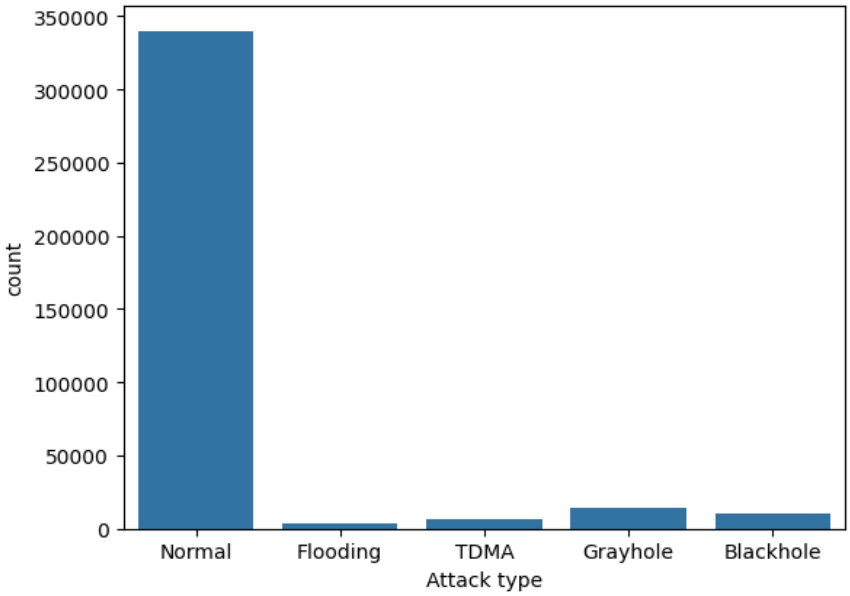


**Figure 2 Class Distribution**

For implementation, WSN-DS dataset is used, which has four major attack classes: flooding, TDMA, Gray hole, Black hole, and one normal class. While the dataset is largely dominated by samples from the Normal class, the attack categories are represented with relatively balanced distributions of instances. The training and testing accuracy of the model are plotted in Fig. 3 correspond to different number of epoch values. The training and test accuracy of the CNN-LSTM model corresponding to different epoch values are shown in **Figure 3**

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

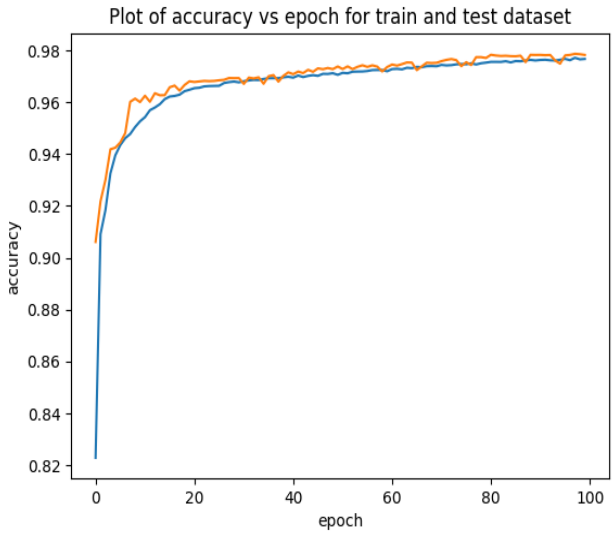


**Figure 3 Accuracy vs epoch**

As shown in **Figure 4**, the model's training and test losses represent the number of epoch values. The model loss is reduced to 2 percent for the model training.

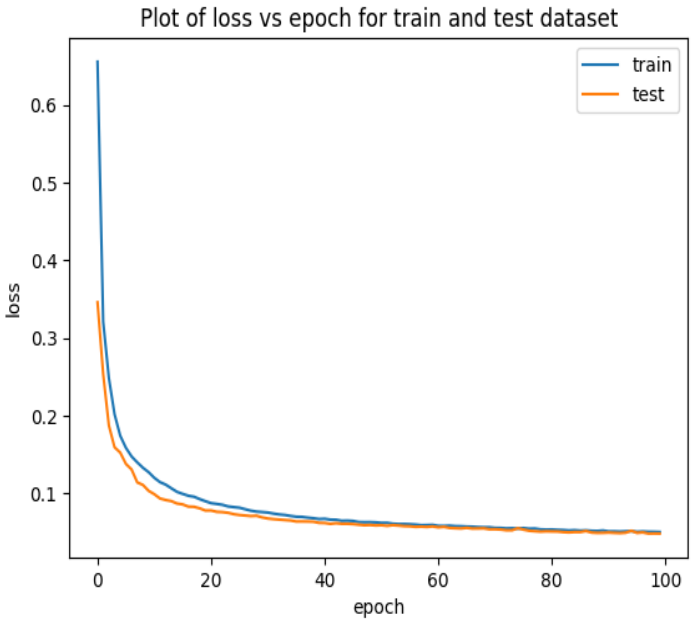


**Figure 4 Training and test model loss**

The classification report presented in **Figure 5** summarizes the results.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

| | precision | recall | f1-score |
|---|---|---|---|
| Normal | 0.66 | 0.91 | 0.76 |
| Flooding | 0.90 | 0.94 | 0.92 |
| TDMA | 0.85 | 0.67 | 0.75 |
| Grayhole | 1.00 | 1.00 | 1.00 |
| Blackhole | 0.93 | 0.92 | 0.92 |
| accuracy | | | 0.98 |
| macro avg | 0.87 | 0.89 | 0.87 |
| weighted avg | 0.98 | 0.98 | 0.98 |

**Figure 5 Classification Report**

The Confusion Matrix, shown in **Figure 6,** is a valuable tool for assessing the effectiveness of a classification model. It presents a comprehensive summary of correct and incorrect predictions across all categories.

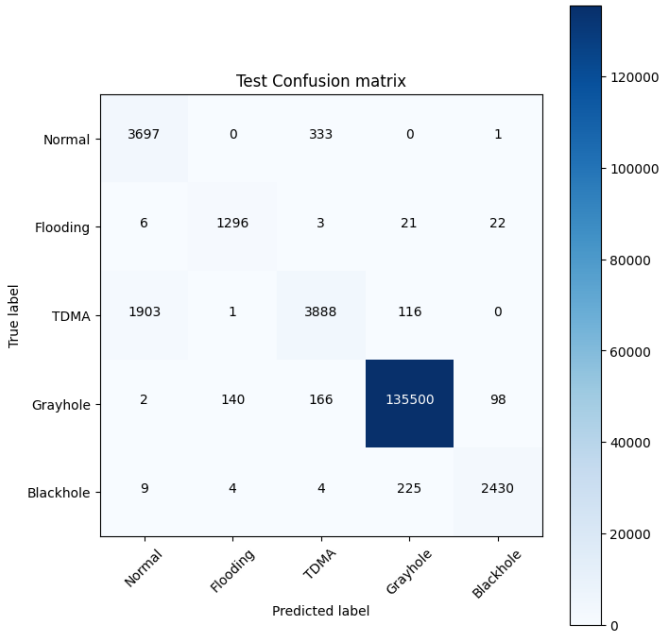


**Figure 6 Confusion Matrix**

As shown in **Figure 7**, the AUC-ROC curve is drawn for each class in the dataset. The x-axis shows the false positive rate and y-axis shows the true positive rate.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

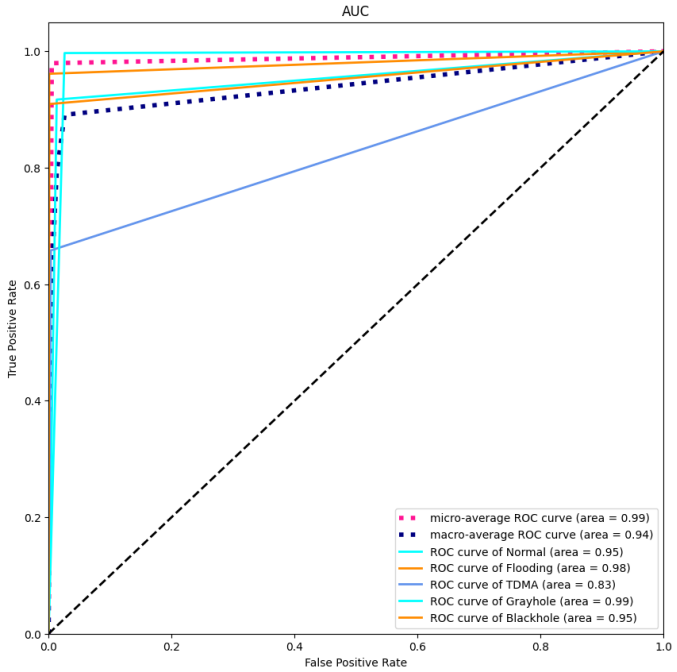


**Figure 7 ROC curve**

### 4.2. Comparative analysis against state-of-the art models

Table 4 compares the proposed framework's performance with current state-of-the-art techniques.

**Table 4 Comparison between proposed model and existing models**

| Ref. | Publication Year | Methodology | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| [18] | 2017 | CNN-RNN | 93.8 | 99.7 | 92.6 | 96.0 |
| [19] | 2018 | RF | High | 98 | 97 | 97 |
| [20] | 2019 | AdaBoost using EFS & SMOTE | 81.83 | 81.83 | 100 | 90.01 |
| [21] | 2020 | CNN-BiLSTM | 83.58 | 85.82 | 84.49 | 85.14 |
| [14] | 2021 | LSTMAE | 97.52 | 96.53 | 97.52 | 97.02 |
| [22] | 2022 | CNN-LSTM | 90.73 | 86.38 | 93.17 | 89.65 |
| [23] | 2023 | DUA-IDS | 88.47 | 87.94 | - | 88.54 |
| [24] | 2024 | GRU-GWO-FS | 96.37 | 73.29 | 64.66 | 68.71 |
| [25] | 2024 | DVA CNN | 96.51 | 93.88 | 88.39 | 90.29 |
| | **Proposed** | **CNN-BiLSTM** | **98** | **98** | **98** | **98** |

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

**This performance analysis and comparison is depicted in the form of clustered column chart in Figure 8.**

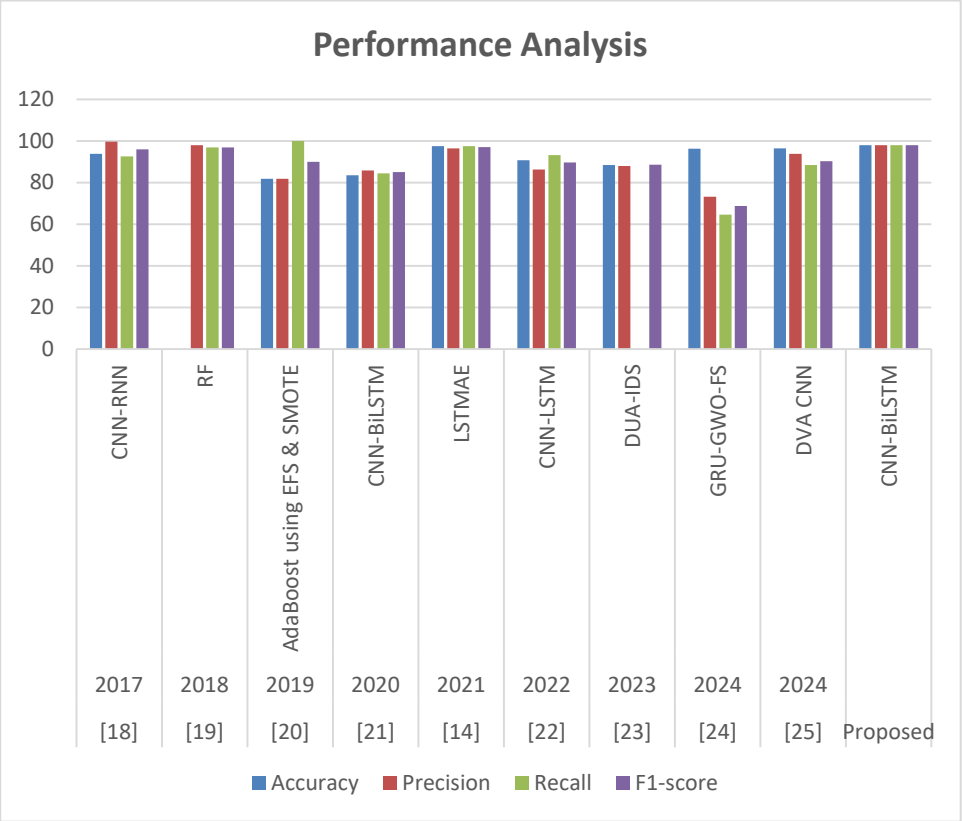


**Figure 8 Performance Analysis**

## 5. CONCLUSION AND FUTURE WORK

With the ever increasing network traffic, the ability to predict future trends with great accuracy in order to manage and optimize network performance is crucial. This paper presents a hybrid deep learning infrastructure, based on the combination of Convolutional Neural Networks (CNN) and Long-Short-term memory (LSTM) network, to enhance the precision of network traffic forecasting. CNN component is a spatial relationship capturing component and LSTM component is a long-term dependency capturing model to detect long-term traffic. The given approach is more precise in making predictions compared to the classic deep learning models since it incorporates the capability of CNN to produce features with the sequencing of LSTM. The proposed traffic classification was experimented in the WSNDS dataset. The performance of the model was assessed using various evaluation metrics such as accuracy, precision, recall and F1-score. The experimental findings showed that this model gives an accuracy rate of 98% which indicates that the model was appropriate and efficient in real time network traffic classification in the wireless sensor networks. Future directions may include to implement the model in a real-world environment to test its responsiveness and optimize it thus using more advanced optimization techniques on the edge devices to achieve low latency inference. Moreover, the model can even be extended to other data sets or types of traffic like IOT or 5G to obtain better performance.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

## REFERENCES

1. Y. Zeng, H. Gu, W. Wei and Y. Guo, "Deep−Full−Range : A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework," in *IEEE Access*, vol. 7, pp. 45182-45190, 2019.
2. Li, Yuchong, and Qinghui Liu, "A comprehensive review study of cyber-attacks and cyber security", Emerging trends and recent developments, Energy Reports 7, pp. 8176-8186, November 2021.
3. W. Wang, M. Zhu, J. Wang, X. Zeng and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks", IEEE International Conference on Intelligence and Security Informatics (ISI), pp.43–48, July 2017.
4. J. Zhang, X. Chen, Y. Xiang, W. Zhou and J. Wu, "Robust network traffic classification", IEEE-ACM Transactions on Networking, pp. 1257–1270, August 2015.
6. D. Ray, "Internet Traffic Classification Using Machine Learning Techniques", 2024.
7. Jadama, A. F., and M. K. Toray, "Ensemble Learning: Methods, Techniques", Application. no., June 2024.
8. Le-Hong, Phuong, Xuan-Hieu Phan, and Tien-Dung Nguyen.: Using dependency analysis to improve question classification. Knowledge and Systems Engineering: Proceedings of the Sixth International Conference KSE 2014. Cham: Springer International Publishing, October 2015.
9. Jouini, Oumayma, et al, "A survey of machine learning in edge computing: Techniques, frameworks, applications, issues, and research directions", Technologies 12.6: 81, June 2024.
10. Kiranyaz, Serkan, et al, "1D convolutional neural networks and applications: A survey", Mechanical systems and signal processing, Volume 151: 107398, April 2021.
11. Almelibari, Alaa A. "Intrusion Detection System Traffic Classification Based on Machine Learning with Correlation-Based Filtering and a Genetic Algorithm-Inspired Feature Selection Method for IoT Networks." *Engineering, Technology & Applied Science Research* 15.5 (2025): 27430-27435.
12. Pinhasov, Ben, et al, "XAI-based detection of adversarial attacks on deep fake detectors, arXiv preprint arXiv:2403.02955, August 2024.
13. R. Ben Said, Z. Sabir and I. Askerzade, "CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection," in *IEEE Access*, vol. 11, pp. 138732-138747, 2023.
14. Almutairi, Yasmeen S., Bader Alhazmi, and Amr A. Munshi, "Network intrusion detection using machine learning techniques", Advances in Science and Technology Research Journal 16.3: 193-206, 2022.
15. Guezzaz, Azidine, et al., "A reliable network intrusion detection approach using decision tree with enhanced data quality", Security and Communication Networks 2021.1: 1230593, 2021.
16. Lin-Huang, Chang, et al., "Application-based online traffic classification with deep learning models on SDN networks", Advances in Technology Innovation 5.4: 216, 2020

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

17. Al-Turaiki, Isra, et al., "Anomaly-Based Network Intrusion Detection Using Bidirectional Long Short Term Memory and Convolutional Neural Network", ISC Int. J. Inf. Secur. 12.3: 37-44, 2020.

18. Xu, Congyuan, et al, "An intrusion detection system using a deep neural network with gated recurrent units", IEEE Access 6: 48697-48707, 2018.

19. N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan and F. Aloul, "Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset," in *IEEE Access*, vol. 10, pp. 6430-6441, 2022.

20. Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani, " Toward generating a new intrusion detection dataset and intrusion traffic characterization", ICISSp 1.2018, 108-116, 2018.

21. Yulianto, Arif, Parman Sukarno, and Novian Anggis Suwastika, " Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset", Journal of Physics: Conference Series, Vol. 1192. IOP Publishing, 2019.

22. K. Jiang, W. Wang, A. Wang and H. Wu, "Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network," in *IEEE Access*, vol. 8, pp. 32464-32476, 2020.

23. A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi and R. Ahmad, "CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System," in *IEEE Access*, vol. 10, pp. 99837-99849, 2022.

24. Xing, Na, et al, "A dynamic intrusion detection system capable of detecting unknown attacks", International Journal of Advanced Computer Science and Applications 14.7, 2023.

25. Elsaid, Shaimaa Ahmed, et al, "Hybrid intrusion detection models based on GWO optimized deep learning", Discover Applied Sciences 6, 10:531, October 2024.

26. Huang, Jia, et al, "Improved intrusion detection based on hybrid deep learning models and federated learning", Sensors 24.12:4002, June 2024.