International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

The Role of AI-Enabled Optimization in Network Traffic Management

Robin Thomas [1], Dr. Anshu Chaturvedi [2], Dr. D.N. Goswami [3]

[1]SOS Computer Science & Applications, Jiwaji University, Gwalior
[2]Professor, Department of Computer Science, MITS Gwalior
[3]Professor, SOS Computer Science & Applications, Jiwaji University, Gwalior

Abstract

Traffic control optimization is a challenging task for various traffic centers around the world and the majority of existing approaches focus only on developing adaptive methods for normal (recurrent) traffic conditions. Artificial Intelligence (AI)-based methods have been widely adopted to predict network traffic, though with low complexity and high efficiency. This study proposes a deep learning-based intrusion detection system (IDS) using a Convolutional Neural Network (CNN) for network traffic analysis on the UNSW-NB15 dataset. The methodology involves comprehensive data preprocessing, including handling missing values, feature encoding, and addressing class imbalance, followed by feature selection using Mutual Information (MI) to enhance classification efficiency. Experimental results demonstrate that the CNN model achieves 99% accuracy, 99.03% precision, 99.86% recall, and a 99% F1-score, outperforming traditional machine learning models such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Logistic Regression. The ROC curve (AUC = 1.00) and confusion matrix further validate its robustness in distinguishing between normal and malicious traffic. However, early-stage overfitting is observed, necessitating further optimization. This research highlights the effectiveness of deep learning for network security, contributing to the development of scalable, real-time AI-driven security frameworks that enhance cyber threat detection and mitigation in dynamic network environments.

Keywords: Network Traffic, Internet of Things (IoT), Intrusion Detection System (IDS), Cybersecurity, Real-time Network Monitoring, Machine learning

1. INTRODUCTION

The rapid digital transformation of industries has led to an unprecedented surge in data generation and exchange. Modern networks, including enterprise infrastructures, cloud-based systems, and telecommunication networks, serve as the foundation of this digital revolution. The proliferation of emerging technologies such as the Internet of Things (IoT), 5G, and edge computing has further intensified network traffic, necessitating efficient management strategies to ensure seamless connectivity, low latency, and high data throughput. As billions of connected devices continuously generate vast amounts of data, traditional network management techniques struggle to keep up with the dynamic and complex nature of modern traffic flows[1].

Network traffic optimization plays a crucial role in maintaining the efficiency, reliability, and security of digital communications[2]. Conventional traffic management approaches rely on static rule-based

methods, predefined configurations, and manual intervention to control congestion, allocate bandwidth, and ensure Quality of Service (QoS). However, these techniques are increasingly becoming insufficient due to the unpredictable nature of network traffic patterns, diverse data transmission requirements, and the rising threat of cyberattacks.

AI-powered network traffic management leverages ML algorithms to analyze, predict, and optimize network performance in real time. By processing vast datasets from network traffic flows, AI-driven solutions can identify anomalies, detect congestion, and dynamically adjust resource allocation to improve overall network efficiency[3]. The need for more adaptive, intelligent, and automated network traffic management solutions has led to the integration of Artificial Intelligence (AI) and Machine Learning (ML) techniques in network optimization. AI-based Network Intrusion Detection Systems (NIDS) can classify normal and malicious network activities, reducing false-positive alerts while enhancing the accuracy of cyber threat detection [4]. Machine learning are widely used to detect traffic anomalies, while deep learning techniques offer more advanced pattern recognition capabilities for real-time security monitoring[5].

1.1 Motivation and Contribution of the paper

The growing complexity and volume of network traffic have made traditional intrusion detection systems (IDS) ineffective against evolving cyber threats, necessitating advanced AI-driven solutions. Machine learning (ML) approaches, particularly Convolutional Neural Networks (CNN), can automatically learn hierarchical patterns in network traffic, enhancing detection accuracy while reducing manual feature engineering. This study aims to develop a robust IDS by integrating preprocessing techniques, feature selection using Mutual Information (MI), and CNN-based classification, ensuring improved performance in detecting both normal and malicious activities. This research seeks to enhance computational efficiency and provide a scalable framework for real-time network security. The key contribution of paper are as:

- Leverages the UNSW-NB15 dataset, incorporating diverse attack scenarios to enhance model generalizability and real-world applicability.
- Implements data balancing techniques missing value handling, duplicate removal, categorical encoding, and class imbalance correction using oversampling techniques.
- Utilizes Mutual Information (MI) to identify significant features, improving classification accuracy and computational efficiency.
- Proposes a CNN-based approach for network intrusion detection, leveraging deep learning's capability to extract complex traffic patterns and improve threat detection accuracy.
- Determine model effectiveness using standard classification metrics, including accuracy, precision, recall, F1-score, and ROC curve.

1.2 Justification and Novelty

The increasing frequency and sophistication of cyber threats necessitate advanced intrusion detection systems (IDS) that can efficiently differentiate between normal and malicious network traffic. Traditional machine learning models often struggle with feature extraction and classification in high-dimensional network data, leading to suboptimal performance. This study justifies the use of CNN by demonstrating their superior ability to automatically learn spatial dependencies in network traffic data, enhancing detection accuracy. The novelty of this work lies in integrating Mutual Information (MI)-based feature

selection with a CNN-based IDS, optimizing computational efficiency while maintaining high detection performance. The comprehensive evaluation using accuracy, precision, recall, F1-score, and ROC curve further validates the model's robustness.

1.3 Structure of the paper

The paper that follows is structured as follows: **Section 2** provides the background study on network traffic analysis using machine learning techniques, proposed methodology and model implementation are present in **Section 3**. **Section 4** provides the Experiment results of the proposed model and a conclusion with future work presented in **Section 5**.

2. LITERATURE REVIEW

This section discusses the state-of-the-art security approaches that use machine learning for network traffic detection in IoT. Various research articles are provide in below:

Sowah *et al.*, (2024) propose two methods for P2P detection, classification and control namely: a logistic regression analysis in WEKA and was used to model the feature selection and detection, and a hybrid system comprising Self Organizing Map (SOM) and Multi-Layer Perceptron (MLP) Neural Network. The logistic regression model detected 98.78% of P2P activity. The hybrid technique could detect 99.93%, 99.75% and 100% P2P activity in 4840, 2001 and 1473 instances of network traffic data respectively. This suggests that when P2P applications take on new features in the future, an unsupervised learning algorithm such as SOM has the potential to identify new P2P features for effective control and management on any given network [6].

Ayodele and Buttigieg, (2024) presents a stack model of Recurrent Neural Networks (RNN), Long-Short-Term memory (LSTM), and Gated Recurrent Networks (GRU) to combat cyberattacks by utilising multilevel traffic analysis at the packet, flow, and session levels. The proposed RNNLSTM-GRU, RNNGRU-LSTM, and LSTMGRU-RNN models are exploited on the same dataset to compare performance. The results show that the models performed well as binary and multiclassifiers, with an accuracy of up to 99.99%. With flow and session-level training, the performance was somewhat lower, achieving accuracies of up to 97.63% [7].

Amos, Narendran and Keerthivasan, (2024) improve Traffic sign recognition using machine learning to improve comprehension of traffic signs. The Novel Artificial Neural Network (ANN) method is compared with Recurrent Neural Network (RNN) to find accuracy. The training and testing splits are varied for each network type. The G Power test used is about 85% with clincalc.com the process is done for α=0.05 and N=0.85 iterations. Novel Artificial Neural Network (ANN) (90.585 %) has the increased accuracy over Recurrent Neural Network (RNN) (76.649 %) with a statistical significance value of 0.042 ($p < 0.05$). Accuracy of Novel Artificial Neural Network (ANN) is significantly better, when compared to accuracy of Recurrent Neural Network (RNN) [8].

Brych *et al.*, (2024) applied deep learning architectures, including CNN and LSTM, to enhance traffic classification and anomaly detection. The results showed that the CNN model achieved a training accuracy of 72% and a validation accuracy of 57%, while the LSTM model achieved a training accuracy of 57% and a validation accuracy of 64%. The steady decrease in the validation loss of the LSTM model highlights its advantages in dealing with sequential data and super-generalization. Both models outperform traditional rule-based systems, which typically achieve an accuracy of 50% to 60%. Future work should focus on optimizing hyperparameters, and hybrid CNN-LSTM models, and applying these

techniques to larger real-world datasets to improve real-time traffic analysis and network management [9].

Qiu, (2023) proposes a network traffic classification method based on Federated learning and Extreme Learning Machine (NTFLELM). NTFLELM does not need to aggregate data for centralized operation, and only needs to pass model parameters, that is, it can use multiple local models to train global models cooperatively, which effectively solves the problems of isolated data island and the lack of labeled data.At the same time, it protects data privacy by sharing only model parameters but not specific data. The NTFLELM is evaluated on the KDD CUP99 data set, and the experimental results show that compared with the benchmark algorithms, the accuracy of NTFLELM network traffic classification is improved by 12% [10].

Fadheel, Al-Mawee and Carr, (2022) examined and assessed the usage of URL Lexical and Network Traffic features to detect malicious URLs. In the study, three methodologies are used: Complete Features, KMO test as a features selection method, and PCA as a dimensionality method, which are tested by LR, SVM, and KNN classification algorithms and evaluated by the Confusion Matrix Accuracy measure. Using Network Traffics features (ISCXURL dataset), the W/O approach: LR, SVM, and KNN has 92%, 94%, and 93% accuracy. The KMO approach: SVM has 91% accuracy. The PCA approach: LR and SVM have 92% and 94% accuracy, surpassing the use of Lexical features (UCI dataset). In contrast, using Lexical features (UCI dataset), the KMO approach: LR and KNN has 90% and 94% accuracy. The PCA approach: KNN has 95% accuracy, surpassing the use of Network Traffic features (ISCXURL dataset) [11].

Koumar and Cejka, (2022) deals with detection of periodic behavioral patterns of the communication that can be detected using time series created from network traffic by autocorrelation function and Lomb-Scargle periodogram. The revealed characteristics of the periodic behavior can be further exploited to recognize particular applications. We have experimented with the created dataset of 61 classes, and trained a machine learning classifier based on XGBoost that performed the best in our experiments, reaching 90% F1-score [12].

Table 1 presents a concise overview of studies on network traffic analysis and cybersecurity, detailing methodologies, datasets, and key findings. It also highlights challenges and potential areas for future research advancements.

TABLE I. SUMMARY OF RECENT STUDIES ON NETWORK TRAFFIC ANALYSIS AND CYBERSECURITY APPROACHES

Study	Approach	Dataset	Performance	Challenges/Future Direction
Sowah et al., 2024	Logistic Regression & Hybrid SOM-MLP	Network Traffic Data	Logistic Regression: 98.78%, Hybrid: 99.93%-100%	Adapting SOM for evolving P2P patterns; improving detection of encrypted P2P traffic.
Ayodele &	Stack RNN	Cyberattack	Up to 99.99%	Handling adversarial

Buttigieg, 2024	Model (RNN-LSTM-GRU)	Traffic Data	(Packet Level), 97.63% (Flow/Session Level)	attacks; optimizing performance on larger datasets.
Amos, Narendran & Keerthivasan, 2024	ANN vs. RNN	Traffic Sign Recognition	ANN: 90.59%, RNN: 76.65%	Improving real-time processing; testing on diverse road conditions.
Brych et al., 2024	CNN & LSTM for Traffic Classification	Network Traffic Data	CNN: 72% (Train), 57% (Validation); LSTM: 57% (Train), 64% (Validation)	Optimizing hyperparameters; exploring hybrid CNN-LSTM models.
Qiu, 2023	Federated Learning & Extreme Learning Machine (NTFLELM)	KDD CUP99	12% higher accuracy than benchmarks	Reducing communication overhead in federated learning; adapting to real-time traffic.
Fadheel, Al-Mawee & Carr, 2022	URL Lexical & Network Traffic Features	ISCXURL, UCI	Network Traffic: 92%-94%; Lexical: 90%-95%	Improving feature selection methods; testing on real-world phishing URLs.
Koumar & Cejka, 2022	Autocorrelation & Lomb-Scargle Periodogram	Custom (61 Classes)	90% F1-Score	Enhancing model interpretability; refining periodic pattern analysis for encrypted traffic.

3. METHODS AND MATERIALS

The proposed methodology integrates Artificial Intelligence (AI) techniques for network traffic through machine learning (ML) approaches. Initially, network traffic data is acquired from publicly available datasets such as UNSW-NB15, ensuring the inclusion of both normal and attack traffic patterns. Preprocessing involves handling missing values, removing duplicate entries, encoding categorical features, and addressing class imbalance using oversampling techniques. Feature selection is performed using Mutual Information (MI) to identify the most significant attributes, thereby improving classification accuracy and computational efficiency. The dataset is then partitioned into training (70%) and testing (30%) subsets to develop and evaluate Convolutional Neural Networks (CNN). Performance evaluation is conducted using standard classification metrics such as accuracy, precision, recall, F1-score, and the Receiver Operating Characteristic (ROC) curve. The structure of proposed methodology is step by step are shows in figure 1.

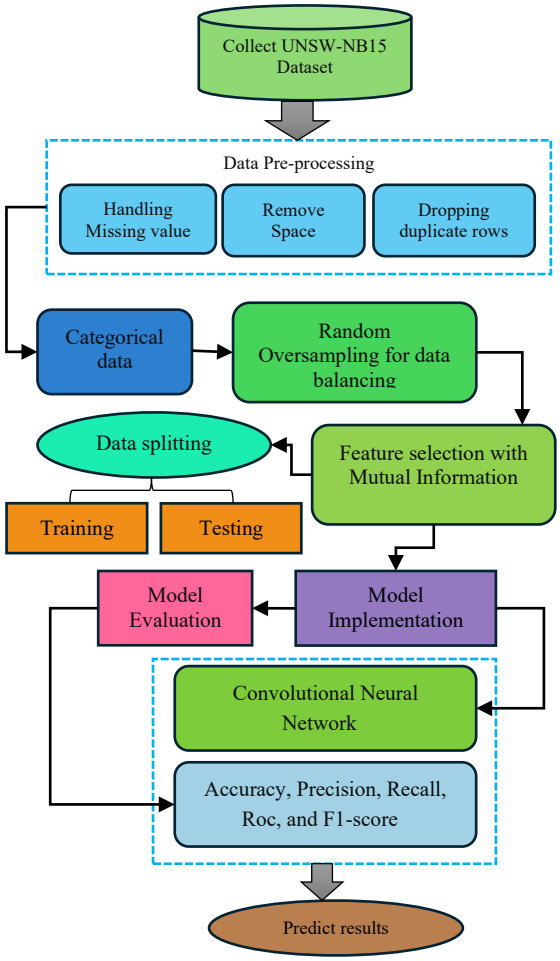

Fig. 1. Flowchart for Network Traffic Analysis

The following steps of the proposed methodology are discussed below:

A. Dataset Collection and EDA

The present study used a widely popular UNSW-NB15 dataset developed at the University of New South Wales (UNSW) in Australia. There are 2.5 million instances present in this data collected in a controlled environment, including both regular, making it relatively large and diverse. The following visualization demonstrates an improved comprehension of the dataset via EDA-based analysis:

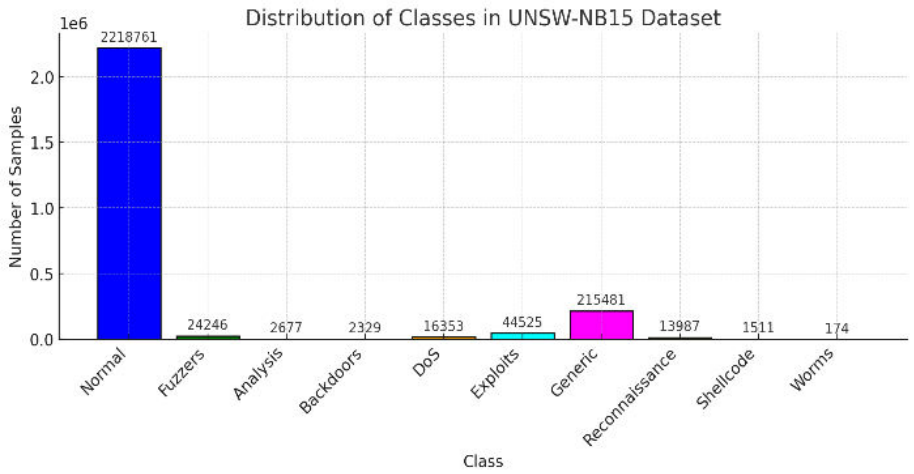

Fig. 2. Bar graph for distribution of classes

The bar chart illustrates the class distribution in the UNSW-NB15 dataset, highlighting a significant imbalance among categories. The "Normal" class dominates with 2,218,761 samples (87.35%), followed by "Generic" attacks with 215,481 samples (8.48%). Other attack categories, such as "Fuzzers" (0.95%), "DoS" (0.64%), and "Exploits" (1.75%), have comparatively fewer instances, while "Shellcode" (0.06%) and "Worms" (0.01%) are the least represented. The variation in sample sizes across classes suggests the need for handling class imbalance when training machine learning models for intrusion detection.

B. Data Pre-Processing

Data preprocessing is a cleaning operation that converts unstructured raw data into a neat, well-structured dataset that may be used for further research. Below is a complete list of the many preprocessing steps:

- **Handling the missing value:** Instances with missing features can be dropped if they comprise a small percentage of the dataset.
- **Removing space:** Use a specific method to replace spaces in column names with no spaces (or underscores) to ensure consistency and avoid issues during analysis.
- **Dropping the duplicate rows:** Identify and remove any rows that are exact duplicates to streamline the dataset and prevent redundancy in analysis.

C. Categorical data

Label encoding is the practice of converting categorical data into numerical values, facilitating their utilization in machine learning algorithms. To train a machine learning model, we must transform categorical values into numerical representations to facilitate the model-building process during the training phase. This is achieved by replacing categorical values with integers ranging from 0 to (n-1), where 'n' represents the total number of unique classes.

D. Random Oversampling for data balancing

Oversampling provides a method to rebalance classes before model training commences. By replicating minority class data points, oversampling balances the playing field and prevents algorithms from disregarding significant yet sparse classes.

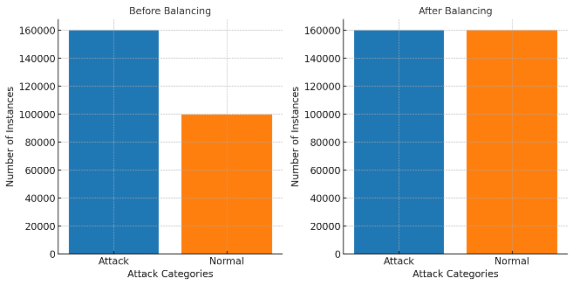

Fig. 3. before and after applying balancing techniques

Figure 3 shows two bar charts comparing the distribution of cases in the Attack and Normal categories. The left plot shows that the Attack category has a greater count (about 160,000) than the Normal category (approximately 100,000), suggesting an imbalance in the dataset. The right plot shows that both categories contain approximately equal instances (about 160,000 each), implying that balancing

techniques (such as oversampling or under-sampling) were used to equalise the distribution. The graphs depict the before-and-after effects of rectifying class inequality.

E. Feature Selection with Mutual Information

Feature selection is a critical step in most classification problems to select an optimal subset of features, increasing the classification accuracy and efficiency [13]. Mutual information underlies the concept of measuring the mutual dependence between two random variables by picking out how much information one of the variables can be obtained from the other one. Mathematically, mutual information can be defined as follows (2):

$$I(X;Y) = \sum_{x,y} p(x,y) log \frac{p(x,y)}{p(x)\ p(y)} \dots\dots\dots\dots\dots() \quad (2)$$

where p(x, y) represents the joint probability distribution function of X and Y, and p(x) and p(y) represent the marginal probability distribution functions of X and Y, respectively. Fig. 4 visual representation of feature score.

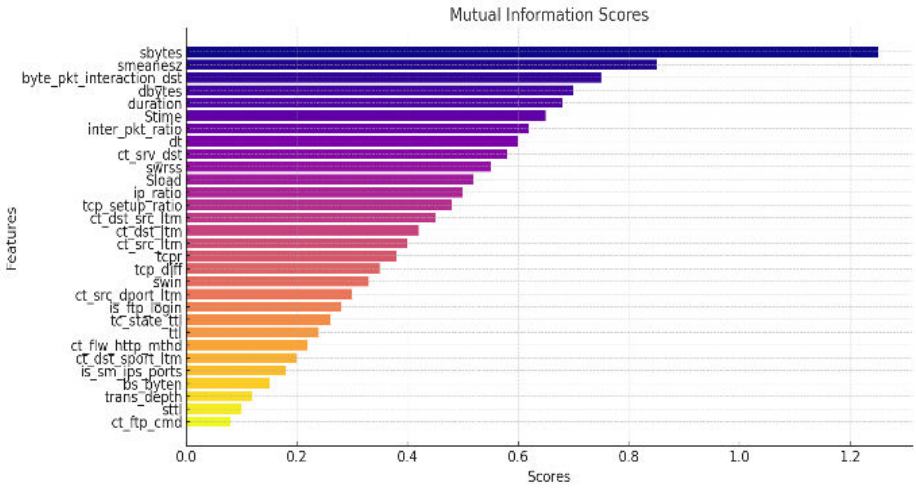

Fig. 4. Bar graph for feature score

Figure 4 shows Mutual Information (MI) scores for dataset features, with sbytes (bytes transferred) as the most informative, followed by means (mean packet size) and others like duration and bytes. Features like ct_ftp_cmd and trans_depth are less relevant. This suggests the dataset is related to network traffic analysis, aiding in tasks like malicious activity detection. The MI scores can guide feature selection to enhance machine learning model performance and reduce overfitting.

F. Train-Test Split

Split the datasets into two parts, namely, training and testing parts. The size of the training dataset is 70% of the entire dataset size, and the remaining 30% is used for testing.

G. Proposed CNN Models

CNNs are considered the basic architecture of deep learning. The architecture of the convolutional neural network has one or more successive convolution layers and a pooling layer. These layers are combined with fully connected and classification layers, respectively [14]. In this study, the CNN model proposed by Kim was used [15]. The architecture of this model is a slight variant of Collobert's CNN architecture[16]. The categories of input data are determined by using these properties[17]. The input

layer consists of n inputs, where each input is represented by a k-dimensional dense vector. Hence, the input x is represented by a $dd\ xx\ kk$ dimensional feature map. Let $x_i \in \mathbb{R}^k$ be the k-dimensional word vector representing the i-th word in the input sentence. A sentence with length n is represented as (2):

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \ldots\ldots (2)$$

Where $\oplus$ is the concatenation operator. A convolution operation involves applying $w \in \mathbb{R}^{hk}$ filter to a window of h words to generate a new feature. For example, using a window of $x_{i:1+h-1}$ words, a new property $ccii$ feature is generated as follows (3):

$$c_i = f\ (w.x_{i:1+h-1} + b \ldots . (3)$$

In Eq. (2), f is a non-linear function such as the hyperbolic tangent and $b \in \mathbb{R}$ is a bias term. A feature map is generated by applying this convolution filter to every possible window of words in the sentence $x_{1:h}, x_{2:h+1}, \ldots x_{n-h+1:n}$. This feature map is generated according to Eq. (4):

$$C = [C_1 C_2, \ldots\ldots C_{n-h+1}] \ldots\ldots\ldots (5)$$

Here $c \in \mathbb{R}^{n-h+1}$. We then take the maximum values corresponding to the filters by applying a max-overtime pooling operation on the feature map. The purpose of this process is to capture the most prominent features in feature maps. The model aims to detect different features using multiple filters in varying window sizes. The outputs of the layer consisting of these features are transferred to the last layer, a fully connected layer. The probability distribution on the labels is calculated in a fully connected softmax layer. The architecture consists of an embedding layer, followed by convolutional layers with ReLU activation, max pooling, and fully connected layers. Hyperparameters are optimized, including a filter size of (3,4,5), 128 filters per layer, a dropout rate of 0.5 to prevent overfitting, and an Adam optimizer with a learning rate of 0.001. The model is trained for 10–20 epochs with a batch size of 64, using categorical cross-entropy as the loss function. Performance is evaluated using accuracy, precision, recall, F1-score, and a confusion matrix.

H. Model Evaluation

Several evaluation methods have been used to evaluate the performance of classifiers. Many metrics can be calculated by using a confusion matrix. confusion matrix, classification report, accuracy, based on recall, f1-score, and precision were the matrices employed. confusion matrix. To determine the classification matrix above the following values were first computed using a confusion matrix. A confusion matrix is defined by the following:

- **True Positive (TP):** Correctly predicted the class as 'positive' when the actual class is also positive.
- **False Positive (FP):** Incorrectly predicted the class as 'positive' when the actual class is negative. It is also called type I error.
- **False Negative (FN):** Correctly predicted the class as 'negative' when the actual class is also negative.
- **True Negative (TN):** Incorrectly predicted the class as 'negative' when the actual class is positive. It is also called a type II error

Accuracy: The easiest measure to understand is accuracy. It is calculated by dividing the total number of occurrences by the number of right guesses and then multiplying the result by 100. show in Equation (5).

$$Accuracy = 100 \times \frac{TP+TN}{N}$$

Precision: Measuring the percentage of samples that are accurately identified for a given class given all predictions for that class is known as precision. Here is its formula (5):

$$Precision = \frac{TP}{TP + FP} \ldots . (5)$$

Recall: The rate at which samples are properly categorized for a certain class type, given all instances of that class type, is known as recall. The formula (6) is used to compute it.

$$\text{Recall} = \frac{TP}{(TP + FN)} \ldots \ldots . (8)$$

F1 score: The precision and recall scores are harmonically averaged to produce the F1 score, also called the F measure. In every case, the F measure will be closer to the lower precision or recall value. Here is a definition of the F1 score (7):

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \ldots \ldots . (9)$$

Loss: Loss represents the error or difference between the predicted output and the actual output. Optimization algorithms use it to update the model weights.

Receiver Operating Characteristics (ROC): The Receiver Operating Characteristics (ROC) curve provides a graphical representation of the classification model performance. The False Positive Rate (FPR) is plotted on the x-axis and the True Positive Rate (TPR) is on the y-axis in the ROC space.

4. RESULT ANALYSIS AND DISCUSSION

The experiments presented in this work are conducted on standard libraries of Python 3.11.1 developed with Jupiter Notebook 6.5.2. Hardware details include Microsoft Windows 11 x64-based-pc, Intel Core i5 processor and 16 GB RAM. The ML models are built, tested, and evaluated on the Scikit Learn ML Python framework. This section provides the proposed model experiment results for network traffic analysis on the UNSW NB15 dataset across the performance matrix, including accuracy, precision, ROC, recall, and f1-score.

TABLE II. FINDINGS OF CNN MODEL FOR NETWORK TRAFFIC ANALYSIS ON UNSW-NB-15 DATASET

Evaluation parameters	Performance (%)
CNN Accuracy	99
CNN Precision	99.03
CNN Recall	99.86
CNN F1-score	99
CNN error rate	0.021

The Convolutional Neural Network (CNN) model exhibits exceptional performance, achieving 99% accuracy, 99.03% precision, 99.86% recall, and an F1-score of 99% shows in table II. The high recall indicates the model's ability to correctly identify positive cases, while the strong precision reflects its effectiveness in minimizing false positives. The balanced F1-score confirms the model's overall efficiency in classification. Furthermore, with an impressively low error rate of 0.021, the CNN demonstrates remarkable reliability and robustness in predictive accuracy.

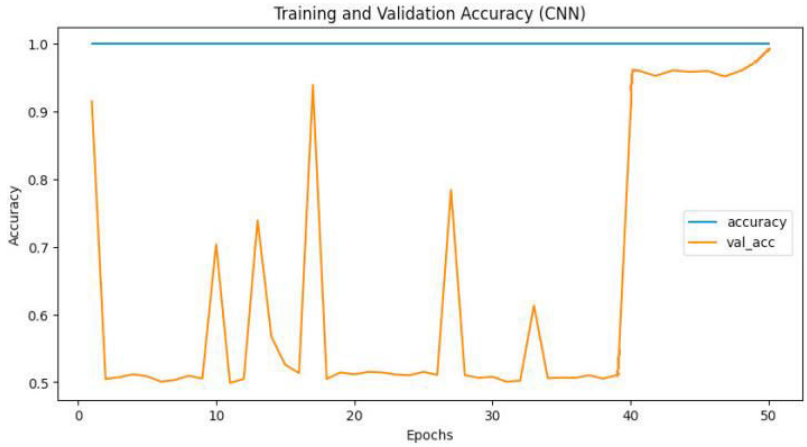

Fig. 5. CNN model training and validation accuracy curve

The graph illustrates the training and validation accuracy of a CNN model over multiple epochs. The training accuracy remains consistently high at around 1.0, while the validation accuracy fluctuates between 0.5 and 0.9, suggesting potential overfitting. However, in later epochs (40-50), validation accuracy stabilizes and approaches training accuracy, indicating improved generalization to unseen data.

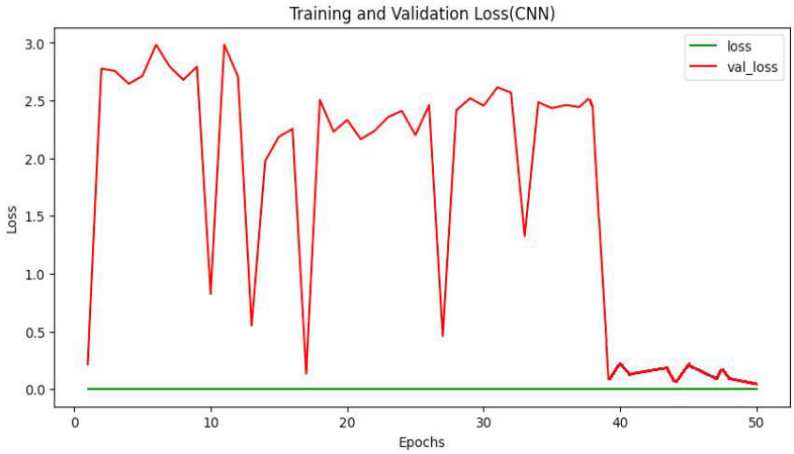

Fig. 6. CNN model training and validation loss curve

The graph depicts the training and validation loss of a CNN model over multiple epochs. The training loss remains low and stable, while the validation loss fluctuates significantly, indicating instability in generalization. However, in later epochs (40-50), the validation loss decreases and stabilizes, suggesting improved generalization and reduced overfitting.

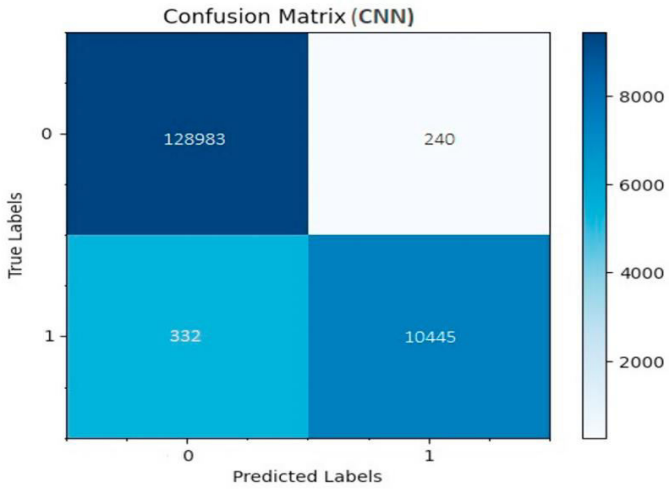

Fig. 7. Confusion matrix for CNN model

The confusion matrix for the CNN model reveals the model's performance in binary classification. The matrix shows that for the negative class (0), the model correctly predicted 128,983 instances and misclassified 240 instances as positive. For the positive class (1), the model correctly identified 10,445 instances while incorrectly classifying 332 instances as negative. The high number of true negatives (128,983) and true positives (10,445) suggests strong overall performance, with relatively low misclassification rates. The color intensity of the matrix, ranging from light to dark blue, visually emphasizes the magnitude of correct and incorrect predictions, highlighting the model's effectiveness in distinguishing between the two classes.

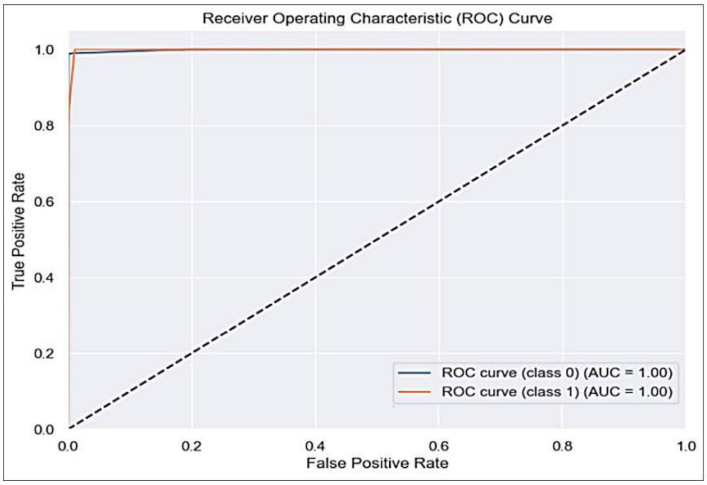

Fig. 8. ROC curve for CNN model

The ROC curve for the CNN model shows outstanding classification performance, with the curve closely hugging the top-left corner. Both classes (0 and 1) achieve an AUC of 1.00, indicating perfect discrimination between positive and negative instances. The significant deviation from the diagonal random classifier line highlights the model's high predictive accuracy and minimal false positive rates.

A. Comparison and discussion

This section compares ML and DL model performance for Network traffic analysis. Table III presents a comparative analysis of various machine learning models based on key performance metrics for network

traffic analysis. The CNN model outperforms all others, achieving the highest accuracy 99%, precision 99.03%, recall 98.86%, and F1-score 99%, demonstrating its superior feature extraction and classification capabilities. LSTM follows with an accuracy of 94.73%, showing strong sequential learning advantages but still lagging behind CNN. Logistic Regression (LR) performs well with 92.8% accuracy, indicating its effectiveness in linear decision boundaries. However, GRU exhibits the lowest performance, with 77% accuracy and an F1-score of 65%, highlighting its limitations in variant detection tasks.

TABLE III. COMPARISON BETWEEN VARIOUS ML MODELS FOR NETWORK TRAFFIC ANALYSIS

Models	Accuracy	Precision	Recall	F1-score
GRU [18]	77	67	75	65
LR[19]	92.8	92.83	92.8	92.8
LSTM[20]	94.73	91.09	92.55	93.3
CNN	**99**	**99.03**	**98.86**	**99**

The proposed CNN-based IDS offers high accuracy (99%), strong feature extraction, and superior classification performance. Its high recall (99.86%) ensures minimal false negatives, crucial for cybersecurity. Mutual Information-based feature selection enhances efficiency, and the ROC curve (AUC = 1.00) confirms its robustness. However, early-stage overfitting and offline evaluation remain challenges. Future improvements will focus on regularization, real-time deployment, and adversarial robustness for enhanced scalability.

5. CONCLUSION AND FUTURE SCOPE

Nowadays a large number of traffic passes through the network. However, the network system is unreliable and has safety issues. Different attacking activities may arise on the network traffic. Again the performance, privacy, latency, and control overhead requirements of real-world networks are the fundamental issues in terms of Network performance analysis. Hence analyzing traffic patterns is a very efficient fact that will help to detect anomalies, monitor network availability, and maximize the performance of the network. This study presents an AI-driven intrusion detection approach utilizing a Convolutional Neural Network (CNN) for network traffic analysis on the UNSW-NB15 dataset. The proposed CNN model achieves exceptional classification performance, demonstrating a 99% accuracy, 99.03% precision, 99.86% recall, and a 99% F1-score, with an impressively low error rate of 0.021. Comparative analysis with other machine learning models, including LSTM, GRU, and Logistic Regression, highlights CNN's effectiveness in feature extraction and classification, significantly outperforming traditional approaches. The evaluation metrics, confusion matrix, and ROC curve (AUC = 1.00) confirm the model's robustness and reliability in network traffic classification. However, the model shows signs of overfitting in early training stages, which later stabilizes, indicating a need for further optimization. Additionally, the study is limited to offline evaluation, and real-time deployment challenges remain unaddressed. Future work will focus on improving generalization through advanced regularization techniques, hybrid deep learning models, and real-time implementation.

REFERENCES

[1] S. Chatterjee, S. Satpathy, and A. Nibedita, "Digital Investigation of Network Traffic Using Machine Learning," *EAI Endorsed Trans. Scalable Inf. Syst.*, 2024, doi: 10.4108/eetsis.4055.

[2] S. Mishra, "Network Traffic Analysis Using Machine Learning Techniques in IoT Networks," *Int.*

J. Softw. Innov., vol. 9, no. 4, pp. 107–123, Jan. 2022, doi: 10.4018/IJSI.289172.

[3] H. A. Ahmed, A. Hameed, and N. Z. Bawany, "Network intrusion detection using oversampling technique and machine learning algorithms," *PeerJ Comput. Sci.*, 2022, doi: 10.7717/PEERJ-CS.820.

[4] P. Vanin *et al.*, "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning," 2022. doi: 10.3390/app122211752.

[5] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Comput. Electr. Eng.*, 2022, doi: 10.1016/j.compeleceng.2022.107810.

[6] R. A. Sowah, G. A. Mills, E. Togo, P. Pomary, and G. Osei, "Efficient Computer Networks Peer-To-Peer (P2P) Traffic Management and Control Using Machine Learning with Open Source Tools," in *2024 IEEE 9th International Conference on Adaptive Science and Technology (ICAST)*, IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/ICAST61769.2024.10856436.

[7] B. Ayodele and V. Buttigieg, "A Multi-Level Network Traffic Classification in Combating Cyberattacks Using Stack Deep Learning Models," in *2024 8th Cyber Security in Networking Conference (CSNet)*, IEEE, Dec. 2024, pp. 143–146. doi: 10.1109/CSNet64211.2024.10851735.

[8] P. Amos, S. Narendran, and M. Keerthivasan, "Analysis Of Traffic Sign Recognition Using Artificial Neural Network Algorithm Compared With Accuracy Of Recurrent Neural Networks," in *2024 9th International Conference on Applying New Technology in Green Buildings (ATiGB)*, IEEE, Aug. 2024, pp. 502–506. doi: 10.1109/ATiGB63471.2024.10717662.

[9] P. Brych, S. Pryshlyak, O. Kovalisko, D. Markiv, G. Boikachov, and M. Brych, "A Study of Traffic Analysis Algorithms in Telecommunication Networks Using Deep Learning to Identify and Classify Data Types," in *2024 IEEE 17th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, 2024, pp. 1–4. doi: 10.1109/TCSET64720.2024.10755752.

[10] C. Qiu, "A Network Traffic Classification Method Based on Federated Learning and Extreme Learning Machine," in *2023 IEEE International Conference on Control, Electronics and Computer Technology, ICCECT 2023*, 2023. doi: 10.1109/ICCECT57938.2023.10140851.

[11] W. Fadheel, W. Al-Mawee, and S. Carr, "On Phishing: URL Lexical and Network Traffic Features Analysis and Knowledge Extraction using Machine Learning Algorithms (A Comparison Study)," in *2022 5th International Conference on Data Science and Information Technology, DSIT 2022 - Proceedings*, 2022. doi: 10.1109/DSIT55514.2022.9943832.

[12] J. Koumar and T. Cejka, "Network traffic classification based on periodic behavior detection," in *Proceedings of the 2022 18th International Conference of Network and Service Management: Intelligent Management of Disruptive Network Technologies and Services, CNSM 2022*, 2022. doi: 10.23919/CNSM55787.2022.9964556.

[13] A. Onan and S. KorukoGlu, "A feature selection model based on genetic rank aggregation for text sentiment classification," *J. Inf. Sci.*, 2017, doi: 10.1177/0165551515613226.

[14] S. Mathur and S. Gupta, "Classification and Detection of Automated Facial Mask to COVID-19 based on Deep CNN Model," in *2023 IEEE 7th Conference on Information and Communication Technology, CICT 2023*, 2023. doi: 10.1109/CICT59886.2023.10455699.

[15] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*,

2014. doi: 10.3115/v1/d14-1181.

[16] D. R. S Masarath, VN Waghmare, S Kumar, RSM Joshitta, "Storage Matched Systems for Single-click Photo Recognitions using CNN," *Int. Conf. Commun. Secur. Artif. Intell. (ICCSAI).*, pp. 1–7, 2024.

[17] S. Nokhwal, P. Chilakalapudi, P. Donekal, S. Nokhwal, S. Pahune, and A. Chaudhary, "Accelerating Neural Network Training: A Brief Review," *ACM Int. Conf. Proceeding Ser.*, pp. 31–35, 2024, doi: 10.1145/3665065.3665071.

[18] S. A. Elsaid, E. Shehab, A. M. Mattar, A. T. Azar, and I. A. Hameed, *Hybrid intrusion detection models based on GWO optimized deep learning*, vol. 6, no. 10. Springer International Publishing, 2024. doi: 10.1007/s42452-024-06209-1.

[19] E. H. Salman, M. A. Taher, Y. I. Hammadi, O. A. Mahmood, A. Muthanna, and A. Koucheryavy, "An Anomaly Intrusion Detection for High-Density Internet of Things Wireless Communication Network Based Deep Learning Algorithms," *Sensors*, vol. 23, no. 1, 2022, doi: 10.3390/s23010206.

[20] Y. N. Rao and K. Suresh Babu, "An Imbalanced Generative Adversarial Network-Based Approach for Network Intrusion Detection in an Imbalanced Dataset," *Sensors*, vol. 23, no. 1, p. 550, Jan. 2023, doi: 10.3390/s23010550.