Volume 12, No.10, October 2025

*Journal of Global Research in Mathematical Archives*

ISSN 2320 – 5822

*UGC Approved Journal*

RESEARCH PAPER

Available online at http://www.jgrma.info

# PERFORMANCE EVALUATION OF ML-BASED ANOMALY DETECTION TECHNIQUES IN NETWORK SECURITY

**Dr. Manish Jain**[1]

[1] Associate Professor, Department of Electronics and Communications, Mandsaur University, Mandsaur (M.P.)
manish.jain@meu.edu.in

**Abstract:** Network security has become a key issue in cybersecurity, and this has forced organizations to protect their valuable resources and sensitive information against dynamic cyber threats. This literature review explains the field's transformation by examining research trends, methodologies, challenges, and achievements that have led to the development of anomaly detection systems. The digital age is evolving rapidly, and one solution to the growing, increasingly advanced cybercrimes is the use of anomaly detection to improve network security. The paper proposes a CNN-based intrusion detection model using the UNSW-NB15 dataset. The used methodology includes a large portion of preprocessing, which includes data cleaning, normalization of Z-scores, Min-max scaling, the selection of features based on the Select-K-Best tool, and balancing of classes by using the SMOTE technique to preserve the quality of data and model strength. The model was trained and evaluated based on an 80/20-percent dataset division, accuracy (ACC), precision (PRE), recall (REC), F1-score (F1), confusion matrix, and ROC-AUC. The experimental results have demonstrated that the CNN model had an accuracy of 98.73% and precision of 96.61% and recall of 96.44, and F1-score of 93.98 with an ROC-AUC of 0.98 and hence good discriminative performance. The proposed CNN model was more reliable and effective than LSTM, Extreme Trees (ET), and Logistic Regression (LR) for detecting network security anomalies.

**Keywords:** Intrusion Detection System (IDS), Network Security, UNSW-NB15 Dataset, Convolutional Neural Network (CNN), Machine Learning.

## 1 INTRODUCTION

One of the primary considerations for information systems in the present digital era is the integrity and confidentiality of information, particularly network security. Decades of augmenting volumes and prevalence of extremely advanced attacks require an effective approach to detect and avert a menace prior to the system being breached [1] [2]. Determining the presence of anomalies is one of the most significant issues in network security because they are considered abnormal behaviors and therefore indicate the possibility of an attack. The former techniques for detecting anomalies relied on manually defined rules that were unable to keep pace with dynamically changing attack patterns. Therefore, the creation of additional auto-adaptive systems of network anomaly detection is of paramount importance. Machine learning can offer an effective solution for anomaly detection, as in this case, no predefined rules are required to implement the strategy. The main reason for performing the detection at a higher level is the determination of the best algorithm and evaluation method, though. In this regard, NB algorithms and cross-validation as an evaluation method were selected because of their ability to handle large volumes of data and to forecast ACC in previous studies.

In the current context, cybersecurity is a worldwide requirement, which is forced by the fact that the protection of systems against undesired, unauthorized, and unexpected interference is of paramount importance [3]. Data breaches, theft of sensitive information, and other risks to system integrity and performance are all examples of this kind of interference. For systems to function smoothly, sensitive data to remain secure, and user confidence to be maintained, prevention against these risks is essential. IDS has long been used as a stronghold of the perimeter defense mechanism.

Anomaly prediction enables proactive prevention of future mistakes and protects sensitive data and critical infrastructure against potential attacks [4]. There are always new obstacles for IT professionals to overcome; study sought to fill that requirement by developing a reliable technique for detecting anomalies in network infrastructures. To maintain and secure network systems, it is essential to detect anomalies. With the ever-changing nature of the Internet and other communication technologies, IT experts are constantly up against new problems. Finding anomalies and responding appropriately while avoiding collateral harm requires the development of sophisticated anomaly-detection systems. Anomalies can manifest in many forms, including irregular traffic patterns, network outages, or unauthorized access. It is common for current systems to fail to classify network data, even though they have obtained good theoretical results [5] [6]. The purpose of this research was to develop an anomaly detection system that is sensitive to network conditions, traffic patterns, and dataset properties. The goal was to create a prototype that might be used in future security systems. This project focuses on a data application with built-in autonomous anomaly detection algorithms, including its deployment and user interface components.

A new approach to detecting anomalies in network data has emerged with the advent of ML. MLs categorization method can discover traffic anomalies. However, conventional machine learning models aren't very good at handling the exponential growth of network traffic in this big data era. Their performance is subpar when dealing with the multi-classification challenge, and their classification ACC is below expectations. So, feature combination selection is now critical for these models [7]. To address partial information loss in network models caused by traffic clipping, this paper introduces a model for network traffic anomaly detection using chaotic neural networks, preprocesses data for multi-information fusion based on network traffic characteristics, and trains a multi-task learning NN classifier.

### 1.1 Motivation and Contribution of Study

Improving network security through more efficient anomaly-detection technologies is an immediate necessity given the proliferation of networked systems and the sophistication of cyberattacks. Protecting sensitive data and critical infrastructure can be challenging with traditional rule-based intrusion detection systems, which are limited in adaptability and unable to keep pace with evolving attack techniques. A potential substitute is anomaly detection methods based on ML, which can learn from data trends and spot outliers without depending on rigid rules. However, challenges such as feature selection, scalability in big data environments, and performance in multi-class scenarios still persist, making it necessary to evaluate and compare different ML approaches[8]. Therefore, the motivation for this study is to conduct a comprehensive performance evaluation of ML-based anomaly detection techniques to identify effective solutions that enhance the reliability, ACC, and robustness of modern network security systems.

- Introduces a comprehensive anomaly detection framework using the UNSW-NB15 dataset, covering data cleaning, normalization, balancing, and feature selection.
- Employs SMOTE to effectively address severe class imbalance, improving detection of minority (non-attack) instances.
- Reduces dimensionality and improves model efficiency by using Select-K-Best feature selection with an ANOVA F-test to keep the 30 most relevant features.
- Implements CNN for robust classification of normal and attack traffic, leveraging optimal hyperplane separation.
- Performs a thorough evaluation of the model's classification efficacy by utilising confusion matrix-based measures (ACC, PRE, REC, F1, and ROC-AUC).

### 1.2 Justification and Novelty

This study is unique because it improves network security anomaly detection by combining a CNN-based framework with robust preprocessing. Unlike prior studies that often rely on raw or minimally processed data, this approach systematically applies correlation analysis, Select-K-Best feature selection, Z-score standardization, Min-Max normalization, and SMOTE-based class balancing to address issues of dimensionality, feature redundancy, and severe class imbalance. This ensures not only higher detection ACC but also better generalization and stability across evaluation metrics. The suggested CNN model is shown to be more efficient and reliable than sophisticated baselines like LSTM, ET, and LR in the comparison study. The study presents an improved, scalable, and more interpretable method for intrusion detection in complex, diverse network data by merging CNNs' margin-maximizing classification with robust preprocessing.

### 1.3 Structure of Paper

The following is the outline of the paper: Review of the literature on anomaly detection was covered in Section 2. The methodology, including the dataset, preprocessing, and proposed CNN architecture, is detailed in Section 3. Section 4 details the outcomes of the experiments, Section 5 compares them to baseline models, and Section 6 wraps up by discussing the potential future applications.

## 2 LITERATURE REVIEW

This section presents a study that uses various ML-based methods to detect network security anomalies. Table 1 summarizes the studies' findings.

Tiwari and Roy *et al.* (2025) presented stacking-based machine learning approach for identifying network traffic anomalies in embedded devices is proposed. It uses a dataset that includes regular and unusual network traffic patterns. To improve detection, the proposed model uses stacking ensemble learning with multiple base classifiers. RF, XGBoost and SGD are selected as basic learners. These are selected because it helps distinguish harmless and benevolent network traffic. By composing such basic learners, a meta-classifier is able to improve the final predictions adding information in the form of the collective output to these basic learners. The stacking ensemble method identifies network anomalies with 94.4% PRE, which is superior than the individual models[9].

Ness *et al.* (2025) Presented a diverse array of machine learning techniques for the detection of anomalies in network traffic and elucidated the manner in which these models address issues such as feature complexity and class imbalance. Classification using support vector machines (SVMs), XGBoost, LightGBM, NB, and Isolation Forest were tested. The study uses a range of ACC, F1, and REC metrics to assess the efficacy of supervised and unsupervised processes, based on extensive research. Isolation Forest

 

suffers from subpar ACC, whereas XGBoost and LightGBM achieved excellent results, with near-perfect ACC in training (1.0) and robust ACC at test (0.85). The paper has also highlighted the strengths and weaknesses of each model, which is vital for applying the models to draw conclusions in the network anomaly detection process [10].

Sharma *et al.* (2024) reported the efficiency of different ML algorithms for detecting these payloads in the system. There are three datasets that are used in the experiments: Hardware in Loop HAI dataset, Gas Pipeline dataset, and ICS cyber-attack dataset. The experimental procedures are performed using Python and a high-performance GPU. Such a cumulative analysis of these experiments reveals that CNN-Dense net and RF algorithms produce the best results with regards to the ACC, PRE, REC, and ROC Curve in all the datasets considered. It should be noted that other algorithms do not differ much when compared to CNN-Dense net and RF. ML algorithm must therefore be selected basing on the data produced by the application system [11].

N and Bhuavneswari *et al.* (2023) The CTU-IoT dataset are a premier IoT-based home automation dataset, and the primary objective of the research was to identify outliers using ML approaches. Using a basic imputer method, the solution reads packet capture (pcap) data from comma-separated value (.csv) files and removes outliers. Then, the Isolation Forest technique is used to label the dataset as normal or abnormal. To address class imbalance, a balanced dataset can be built using oversampling methods such as SMOTE and Borderline-SMOTE. This work includes all three datasets, the original imbalanced, the SMOTE balanced or the Borderline-SMOTE. These datasets are used to evaluate the performance of several ML algorithms, like SVM, NB, DT, RF among others; the key measures of their performance, like ACC, PRE, REC, F1, and computation time are used [12].

Gulhare *et al.* (2022) proposed a group ML-based anomaly detection algorithm for IoT devices using the Mean-Shift and Local Outlier Factor (LOF) algorithms. The model is evaluated using datasets that are based on the UNSW-NB15 IoT network. Categorization of common or abnormal actions is performed using the ensemble ML method, whereas clustering is performed using the Mean-Shift clustering strategy and OLOF. The suggested model is tested and compared using precision (P), recall (R), accuracy (ACC) and F1-score [13].

Singh and Srivastav *et al.* (2021). The IDS can identify and log different types of attacks —known, unknown, and zero-day—using unsupervised machine learning. The device was based on the OCSVM, and active learning was implemented to identify potential risks before they occurred. To this end, we compared the framework's performance on the UNSW-NB15 and KDD Cup 99 datasets with its performance on the CIC-IDS2017 dataset. According to the outcomes, this framework appears to be doing better than the competition [14].

A number of studies have investigated anomaly detection in IoT and network data using various ensemble and machine learning approaches, yet substantial knowledge gaps remain. Existing approaches often face challenges such as class imbalance, overfitting to specific datasets, high computational complexity, and limited generalizability across heterogeneous IoT environments. While methods such as stacking ensembles, CNN-DenseNet, and LightGBM demonstrate strong performance, their reliance on specific datasets (e.g., CTU-IoT, UNSW-NB15, CIC-IDS2017) limits scalability for real-time applications. Moreover, many models are evaluated in offline or controlled environments, leaving a gap in practical deployment under dynamic network conditions. Therefore, there is a need for lightweight, adaptive, and hybrid frameworks that integrate supervised and unsupervised techniques, robustly address imbalance, and ensure real-time detection of evolving and zero-day attacks in embedded and IoT systems.

**Table 1:** Comparative Analysis of Anomaly Detection Techniques in Network Security

| Author | Methodology | Data | Key Findings | Limitation | Future Work |
|---|---|---|---|---|---|
| Roy, Tiwari and Roy (2025) | Stacking ensemble with RF, XGBoost, and SGD as base learners, meta-classifier for refinement | IoT network traffic dataset (regular & anomalous patterns) | Stacking ensemble achieved 94.4% ACC, outperforming individual models | Limited to selected base classifiers, may not generalize well to unseen traffic patterns | Explore deep learning ensembles and real-time anomaly detection |
| Ness et al. (2025) | Comparison of supervised & unsupervised ML models (Isolation Forest, Naive Bayes, XGBoost, LightGBM, SVM) | Network anomaly dataset (unspecified) | LightGBM outperformed Isolation Forest, with a training ACC of 1.0 and a test ACC of 0.85. | Possible overfitting in LightGBM, dataset imbalance issues | Hybrid models combining supervised & unsupervised learning |
| Sharma, Bajaj and Sahu (2024) | CNN-DenseNet and Random Forest evaluated alongside other ML models | HAI dataset, Gas Pipeline dataset, ICS cyber-attack dataset | CNN-DenseNet and RF provided highest ACC, PRE, REC, ROC across datasets | High computational cost (GPU required), dataset-specific results | Expand experiments with lightweight DL models for embedded IoT |

| | | | | | |
|---|---|---|---|---|---|
| N and Bhuavneswari (2023) | Isolation Forest + data balancing (SMOTE, Borderline-SMOTE) + ML models (SVM, NB, DT, RF) | CTU-IoT dataset (IoT home automation) | Balanced datasets improved results; RF and SVM performed best with oversampling | Class imbalance issue still partially persists; computation time higher after oversampling | Extend framework to real-time IoT traffic with streaming ML |
| Gulhare, Badholia and Sharma (2022) | Use of Mean-Shift clustering in an ensemble setting along with a local outlier factor (LOF) | UNSW-NB15 IoT dataset | The model performed above average in all four metrics: F1, ACC, PRE, and REC. | Only available on the UNSW-NB15 dataset; settings determine clustering performance. | Extend to multi-IoT datasets and optimize cluster selection |
| Singh and Srivastav (2021) | One-Class SVM with Active Learning for unsupervised anomaly detection | CIC-IDS2017, UNSW-NB15, KDD Cup'99 | Framework effectively detected known, unknown, and zero-day attacks; better than baselines | Computational complexity and scalability concerns | Apply deep one-class learning and adaptive active learning |

## 3 METHODOLOGY

Utilizing the UNSW-NB15 dataset, a methodical procedure for anomaly detection in network security is established through data preparation, feature optimization, and model validation. Raw data traffic is initially cleaned, normalized, and standardized to provide consistency and better predictive power. To correct severe class imbalance between the attack and non-attack records, the SMOTE algorithm is used, and to reduce the number of features, Select-K-best feature selection is applied, and 30 most relevant attributes are retained. This data is subsequently partitioned into a state of 80/20 to guarantee an effective training and an objective judgment. A CNN is taught to create an ideal hyperplane that can distinguish between benign and malicious traffic, and the models' performance is assessed by utilizing common metrics like as ACC, PRE, REC, and F1, which are derived from the confusion matrix. Also, ROC-AUC analysis is a good way to gauge the model's discriminatory power because it captures the trade-off between specificity and sensitivity to a sufficient degree. Such extensive approach provides the balance in data representation, dimensionality reduction and the rigor of performance evaluation to the intrusion detection research best practices.

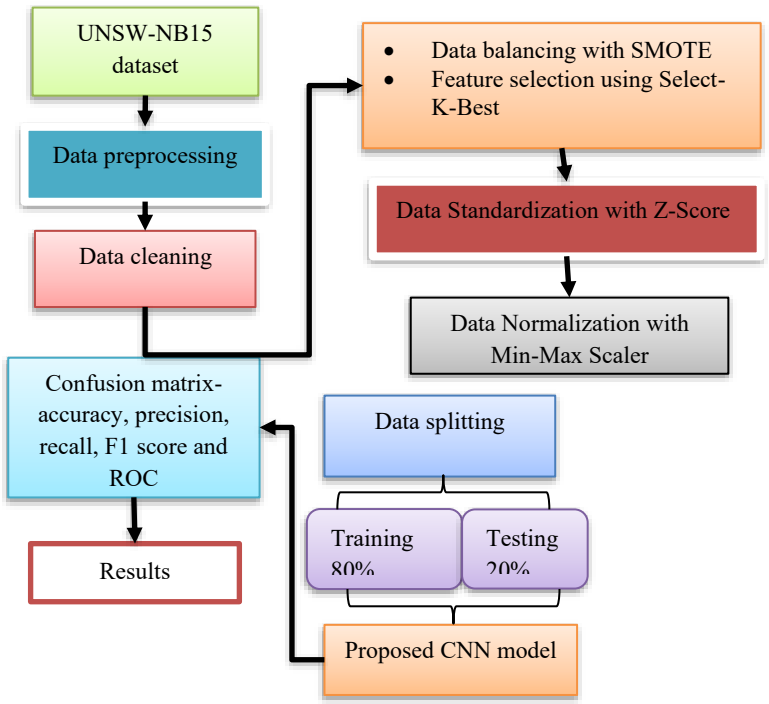


**Figure 1:** Flowchart of Anomaly Detection Techniques in Network Security

The description of each step is presented in the following sections, which also appear in methodology and proposed flowchart:

### 3.1 Data Collection

The Australian Centre for Cyber Security (ACCS) at UNSW used the Perfect-Storm program to generate a database called UNSW-NB15. It contains 100 GB of TCP-Dump-captured raw network data. The result is 2,540,044 records that include varied protocols

 

including TCP, UDP, ICMP, etc. Separate training and testing sets contain 37,000 normal records and 56,000 normal records, respectively, coupled with a variety of attack occurrences, making the dataset ideal for study on intrusion detection.

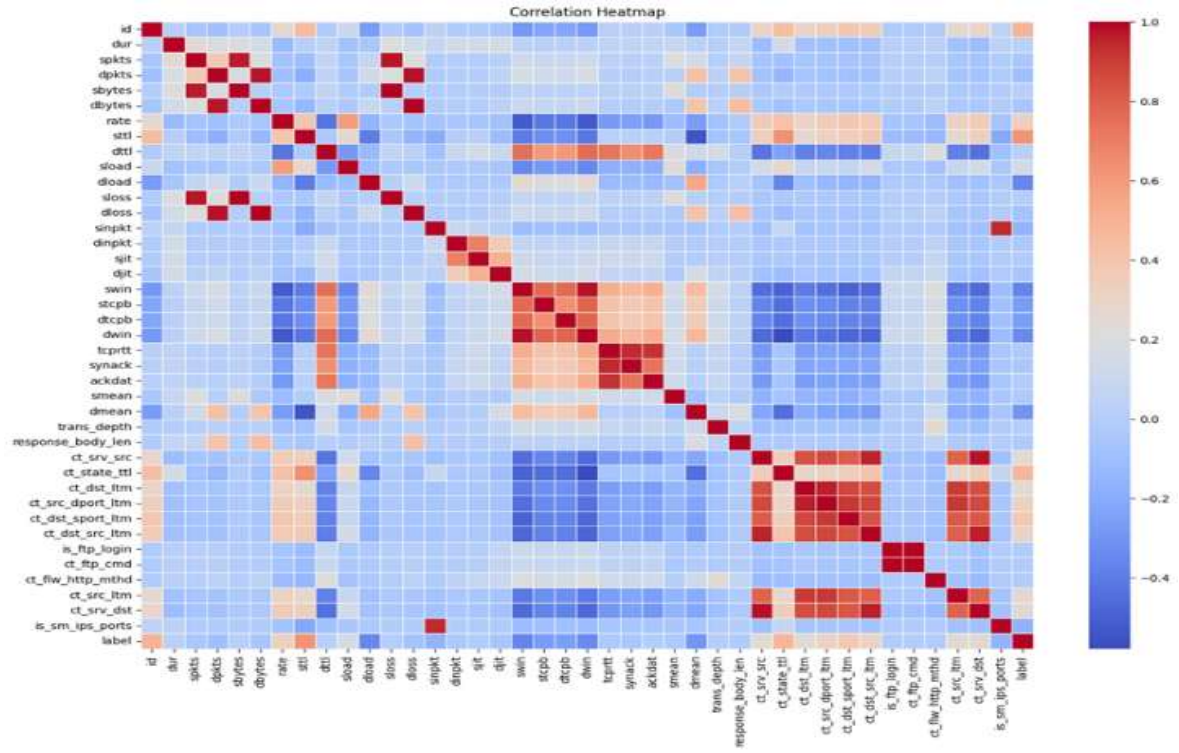


**Figure 2:** Correlation Heatmap

The pairwise correlation coefficients between several dataset features are shown in Figure 2. Color saturation and hue reveal the nature and direction of the relationship. If two variables are highly correlated, then an increase in one cause an increase in the other; conversely, a drop in one cause an increase in the other; and a strong positive correlation shown by red. The diagonal line is entirely red, as each variable is perfectly correlated with itself. The symmetrical nature of the heatmap reflects that the correlation between variable A and B is the same as between B and A. Notably, the 'label' feature, which likely represents the class (e.g., attack vs. non-attack), shows varying degrees of correlation with other features, such as 'smean', 'dmean', and 'ct_srv_dst_ltm'. The detection of strongly-correlated features is a key step in data analysis, which can be used to select better features to achieve better performance and to avoid multicollinearity between features.

### 3.2 Data Preprocessing

Data processing is a vital step towards using the data effectively to enhance the model ACC. It consists of error cleaning, dealing with missing values, standardization of features (z-score normalization), values normalization (Min-Max scaling), class balancing (e.g. SMOTE). These make sure that data is of quality, contribute equally in feature provision, and is less biased, producing more sound predictive models.

### 3.3 Data Cleaning

The primary goal of the process is to improve data quality by identifying errors and inconsistencies. Common data cleanup activities include addressing incomplete data, deleting duplicates, and correcting errors and irrelevant data records.

### 3.4 Data balancing with SMOTE

The SMOTE is used to eliminate overfitting in the study. As an oversampling technique, SMOTE generates synthetic samples from minority populations.

The bar graph shows the counts of two types: Non-Attack and Attack. The number of instances per class is depicted by the y-axis. A count of about 58,000 is shown in the bar labelled non-attack as opposed to the significantly high count of 120,000 shown in the bar labelled Attack. This stark difference in the counts reflects a high degree of class imbalance the "Attack" class being the majority class and the "non-attack" class the minority class. This imbalance is a typical issue of machine learning data and may cause discrimination of a model towards the majority of the data and have low efficiency on the minority data. Such visualization thus makes the balancing techniques absolutely essential prior to training a model.

 

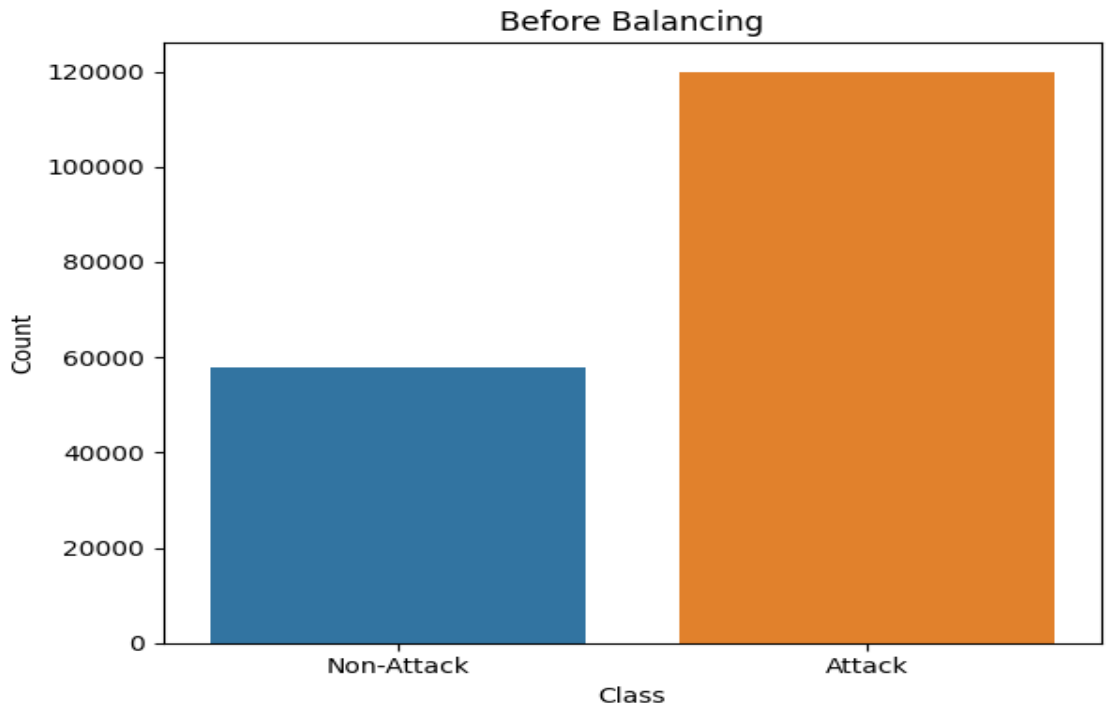


**Figure 3:** Count plot for data distribution

### 3.5 Feature selection using Select-K-Best

Reduce dimensionality with Select-K-Best transformation based on ANOVA F-test to improve model performance. It gives each feature a score based on how well it correlates with the target variable. Consideration of the characteristic for the target variable generally leads to a higher score. By setting the value k to 30, ensured that just the 30 most notable features were kept. The names of the selected features were then extracted from the original dataset, excluding the target variable, using get_support. Additionally, feature importance scores are analyzed to validate the selected features are shows in figure 4.

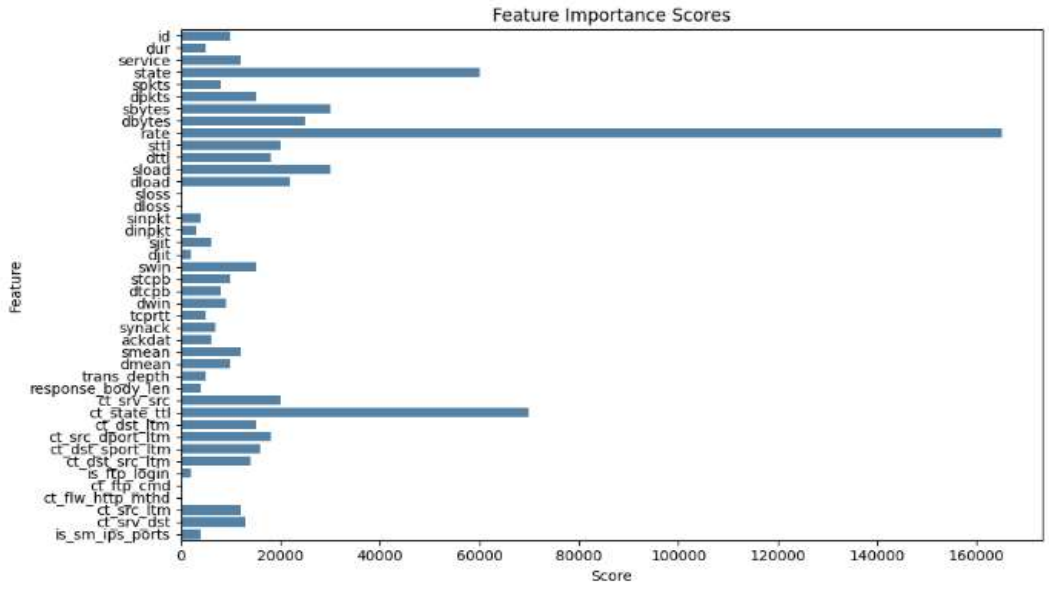


**Figure 4:** Feature importance

Figure 4 shows the weights assigned to different features in an ML model, most likely utilized for a classification job. Each horizontal bar's length corresponds to a score, illustrating how the feature affects the model's predictive capacity. The feature 'rate' stands out with its impressive score, indicating its prominence as the model's decision-making variable. The properties 'ct_state_ttl','state', 'id', and 'ct_dst_ltm' are also highly significant. In contrast, features like 'spkts', 'dpkts', 'sload', 'dload', and 'is_sm_ips_ports' have relatively low scores, indicating they are less relevant for the model's predictions. The determination of feature importance is essential in the application of machine learning because it can inform activities related to feature engineering,

 

improve interpretability (of the model), and potentially simplify the model, potentially reducing dimensionality and the computational burden of the model.

### 3.6 Data Standardization with Z-Score

Standardization (or z-scoring, also called z-scale or z-normalization) is an important technique of feature scaling. It is the act of taking the mean of each of the features and dividing it by the standard deviations of that features value. This methodology is applicable where the input data displays an extensive gap in the feature values. Each characteristic has a consistent scale after standardization, with a mean (μ) of 0 and a standard deviation (σ) of 1. The predictive models' ACC is greatly enhanced by this method. Find the Z-score normalization equation (Equation 1) up there in the mathematical formulation.

$$x_{new} = \frac{x-\mu}{\sigma} \tag{1}$$

The feature's initial value is denoted by x, its standardized form by $x_{new}$, the mean by μ, and the standard deviation by σ.

### 3.7 Data Normalization with Min-Max Scaler

Data features can be scaled to a common value by normalization, a scaling approach. In UNSW-NB 15, the continuous feature's minimum and maximum value ranges are drastically different. Each feature's value range is uniformly linearly mapped in the range of 0 and 1, and the mathematical processing is made easier with the use of a Min-Max scaling method. It is possible to describe the Min-Max scaling technique as

$$X_{scal} = \frac{X-X_{min}}{X_{max}-X_{min}} \tag{2}$$

Scaling data points are represented by $X_{scal}$, while the minimum and maximum values of the input feature X are $X_{min}$ and $X_{max}$, respectively.

### 3.8 Data balancing After SMOTE

A small number of minority-class points inside the domain of majority-class points causes the formation of bridges between those locations. An improved version of SMOTE called Borderline SMOTE can fix this issue with SMOTE. Because of the modest number of over-sampled cases along the orderline, Borderline-SMOTE is able to generate synthetic data within the class choice boundary. Classification tasks necessitate that most techniques learn the borderline for each class during training so that they can do better.

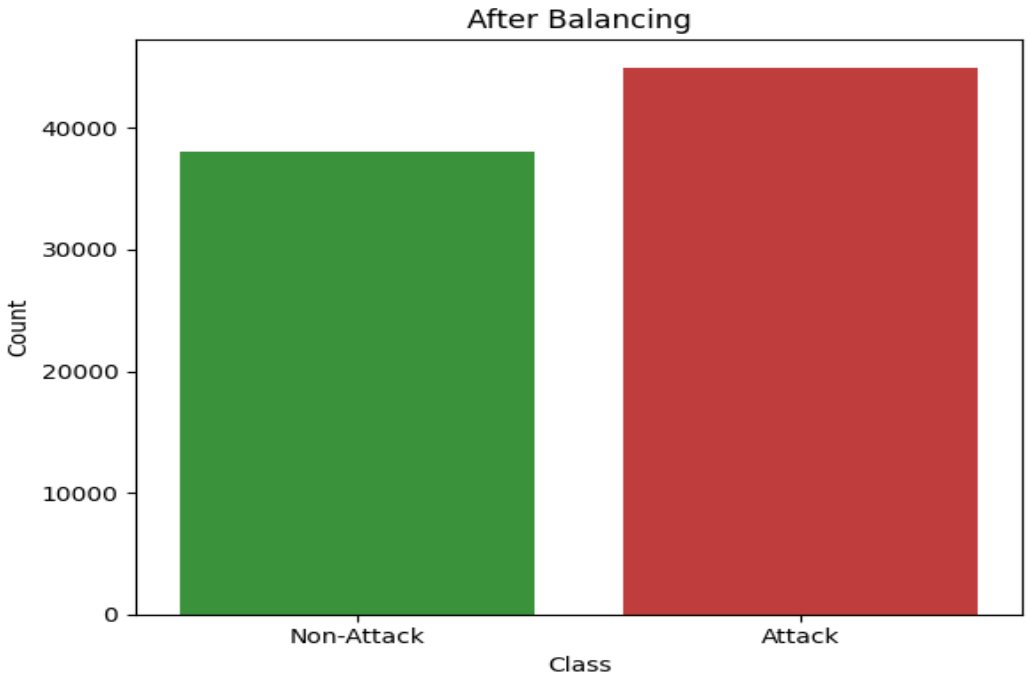


**Figure 5:** Count Plot for data distribution after Balancing

Figure 5 illustrates the distribution of the dataset's classes following the application of a balancing technique. Unlike the previous, imbalanced distribution, this chart shows that the counts for the "Non-Attack" and "Attack" classes are now nearly equal. The count

 

for the "non-attack" class is approximately 38,000, while the "Attack" class has a count of about 44,000. This near-equal distribution suggests that an oversampling technique, such as SMOTE, or an undersampling method was used to adjust the class frequencies. Reducing the likelihood of a machine learning model being biassed towards the initial majority class through the creation of a more balanced dataset can increase performance and generalization, especially when it comes to detecting the minority class.

### 3.9 Data Splitting

The training set encompasses 80% of the data utilized for model training in an 80/20 split, whereas the testing set contains 20% of the data retained for testing. This ensures sufficient data for learning while retaining a portion for unbiased performance evaluation.

### 3.10 CNN Model

CNNs are the image algorithms that are used the most. picture classification, picture recognition, object detection, and other tasks often use convolutional neural networks [15]. Classifying pictures is the DL job where CNN are used the most. Also, CNN is an image dataset with a lot of dimensions; each image has many thousands of pixels, which makes it difficult and big. CNN can figure out the topological features of an image because it is a feed-forward network. CNNs are models that use multiple layers of perceptron's. CNNs use three different layers that are more like neural networks: convolutional, pooling, and fully linked. Each layer has its own job to do. Features extracted from the convolutional layer have made use of it. The current entry's category is determined by the pulled function in the fully linked layer. A pooling layer's function is to decrease the quantity of feature mappings and network parameters.

- **Convolutional Layer:** The feature maps of convolutional layers are built by applying the local connection concept and the weight distribution theory. Through the utilization of location-independent local picture statistics and the highly-connected local pixel neighborhoods, as well as weight distribution goals, the number of parameters required is reduced.
- **Pooling layer:** The pooling layer combines the characteristics of two or more conceptually related convolutional layers by use of a subsampling procedure. A pooling layer entity map (convolutional) unit takes a local patch as input and uses it to get the maximum or average patch value at the output. Later on, with a smaller representation and better robustness, the number of parameters needed drops.
- **Fully connected layer:** The classical multilayer perceptron neural network has fully connected units in every other layer.

A CNN's efficiency and efficacy are greatly affected by how its physical area is organised. Specific activities can be accomplished more swiftly and accurately depending on the construction method, materials used, and arrangement of the layers.

### 3.11 Performance Matrix

One important method for testing the efficacy of ML classification is the Confusion Matrix. Tables showing expected and actual findings can take one of four forms: TP, FN, or TN. The ACC, PRE, REC, and F1 metrics are highly dependent on this matrix.

***Accuracy:*** One of the performance metrics is ACC, which measures how many out of a total number of observations were correctly predicted. Equation (3).

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \tag{3}$$

***Precision:*** The predictive ACC is the ratio of the expected number of positive values to the number of positive values actually counted: Equation (4).

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4}$$

***Recall:*** The percentage of positive accumulator occurrences for which a valid prediction was made is called REC. Equation (5).

$$\text{Recall} = \frac{TP}{TP+FN} \tag{5}$$

***F1_Score:*** An F1 is a harmonic mean of two classification metrics: REC and PRE. Whereas in equation (6).

$$F1 - score = 2 * \frac{(precision*recall}{(precision+recall)} \tag{6}$$

***ROC:*** The ROC curve is a two-dimensional graph that illustrates the trade-off between sensitivity and false-positive rate as the threshold is varied. A classifier's performance can be assessed using it. This statistic assesses the model's capacity to differentiate between classes; a value of 1 indicates excellent discrimination, whilst a value of 0.5 indicates haphazard performance. As a result, ROC-AUC is the go-to statistic for selecting the best classifier.

 

## 4 RESULTS AND DISCUSSION

Analysis of the experimental results is carried out in this section. Performance is reported under some of the metrics such as REC PRE, ACC, and F1. The computer configuration employed in this experiment was an i9-9820X that had a 3.30 GHz clock speed, 2 TB of RAM and an Ubuntu 20.04.1 LTS system. Anaconda Jupiter Notebook Python code was used to develop Python code The results presented in Table 2 depict that the CNN model is effective to detect anomalies. The model registered a good ACC of 98.73%, demonstrating its reliability in general since only two instances of the normal and the attack were misclassified. At the PRE of 96.61%, the CNN performs well as far as false positives are concerned; it does not report many false attacks because the maximum found were only 34 attacks that are indicated incorrectly. The REC measure of 96.44%w also means that the model can accurately target the occurrence of actual attacks, hence eliminating false negative values. In addition, with a F1- score of 93.98, the model is characterized as consistently precise and recalling. These findings or results coupled together determine that the CNN model is a solid and reliable intrusion detection algorithm.

**Table 2:** Performance Parameters for CNN model

| Measures | CNN |
|---|---|
| Accuracy | 98.73 |
| Precision | 96.61 |
| Recall | 96.44 |
| F1-score | 93.98 |

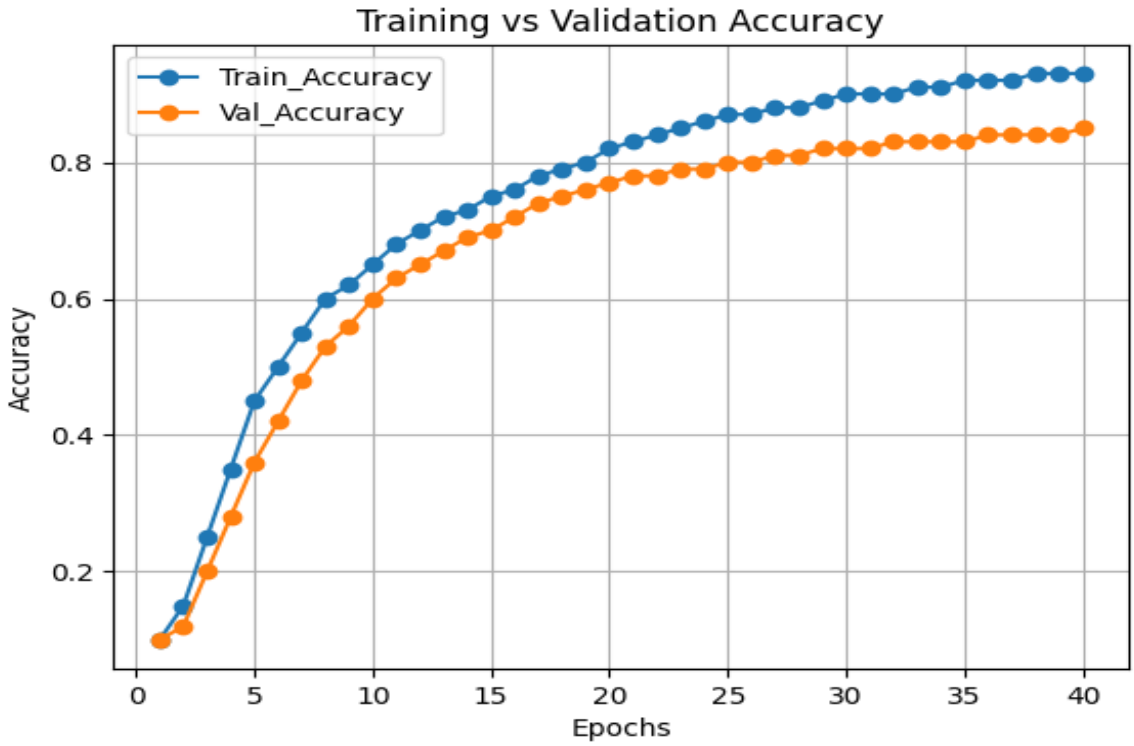


**Figure 6:** Accuracy curve of CNN Model

In Figure 6, the ACC curves for training and validation over 40 epochs show a continuous increase in both, suggesting that the CNN model is improving in performance as the number of epochs increases. The initially training and validation accuracies increase rapidly, which means that the model is quickly grasping the main tendencies in the data early in the training phase. As the epochs advance, the curves curve closer and closer until they are finally stationary, with the training ACC nearly equal to 0.92 and at validation ACC of around 0.85, which shows that there is not much overfitting of the model. The distance between the two curves, however, is not very large, showing that the CNN model retains a good level of learning aptitude whilst being robust on unseen validation data. This overall score proves that the model is effective at finding a balance between learning and generalization.

Figure 7 documents the training and validation losses of the CNN model throughout 40 epochs and clearly shows that both losses keep decreasing as the training advances. The initial difference shows the training loss as greater than 1.0 and the validation loss is a little bit lesser at about 0.8 but both decrease quite fast at the early epochs signifying effective learning. With the passing of epochs, training loss steadily declines, currently standing at about 0.28, whereas the validation loss almost stops changing, being about 0.38, which indicates that the model is well-optimized. The strong congruence between training and validation loss patterns, with no prominent divergence, indicates that no overfitting of the CNN model emerges, and that the model learns well based on the provided information. This consistent reduction in loss validates the robustness and stability of the model in minimizing classification errors.

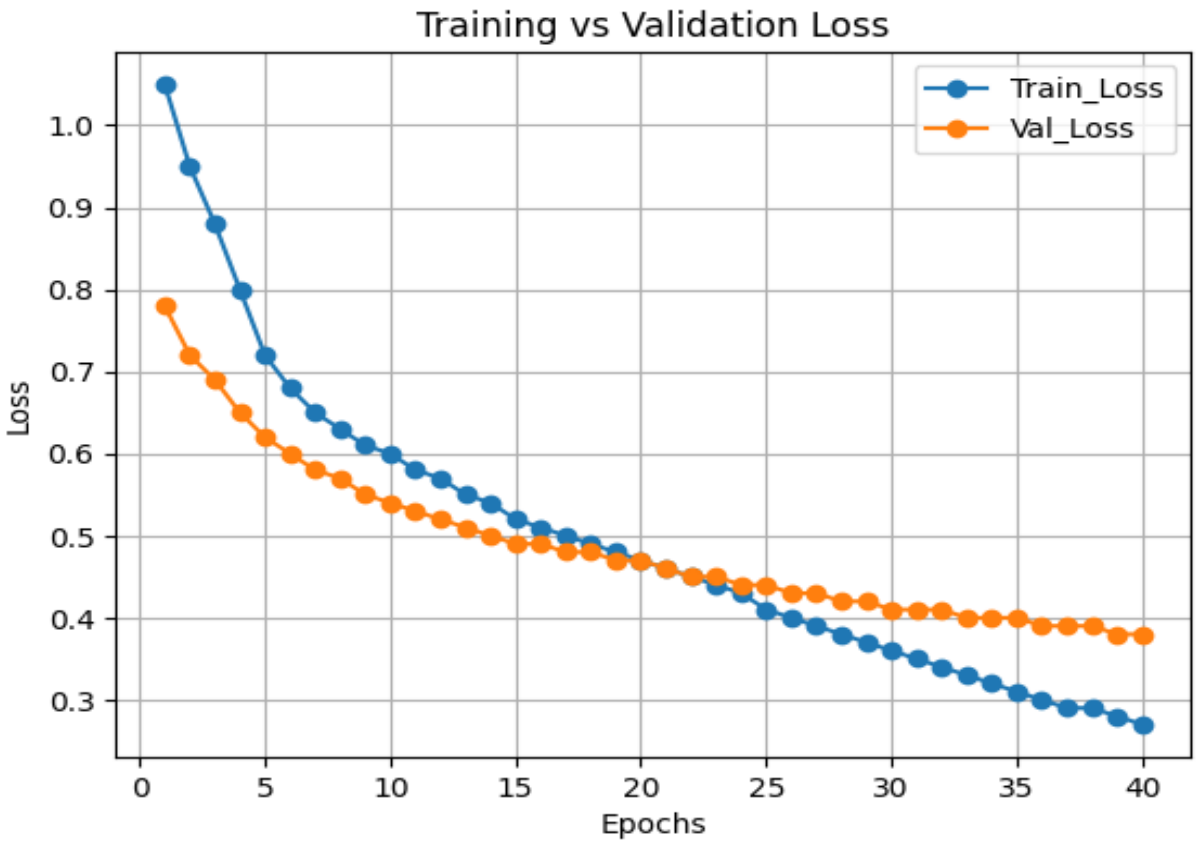


**Figure 7:** loss curve of CNN Model


**Figure 8:** Confusion matrix of CNN model

A CNN model's confusion matrix, shown in Figure 8, would most likely be employed for an intrusion detection or other cybersecurity-related activity, labeling instances as "Attack" or "Non-Attack:". In the given matrix, and show that the model accurately classified 80% of the occurrences as Attack (True Positives) and 80% as Non-Attack (True Negatives). This indicates a strong diagonal, representing correct classifications. Conversely, it incorrectly classified 20% of non-attack instances as Attack (False Positives) and 20% of Attack instances as Non-Attack (False Negatives). The model's robustness is demonstrated by its low FP and FN rates, as well as its high TN and TPR.

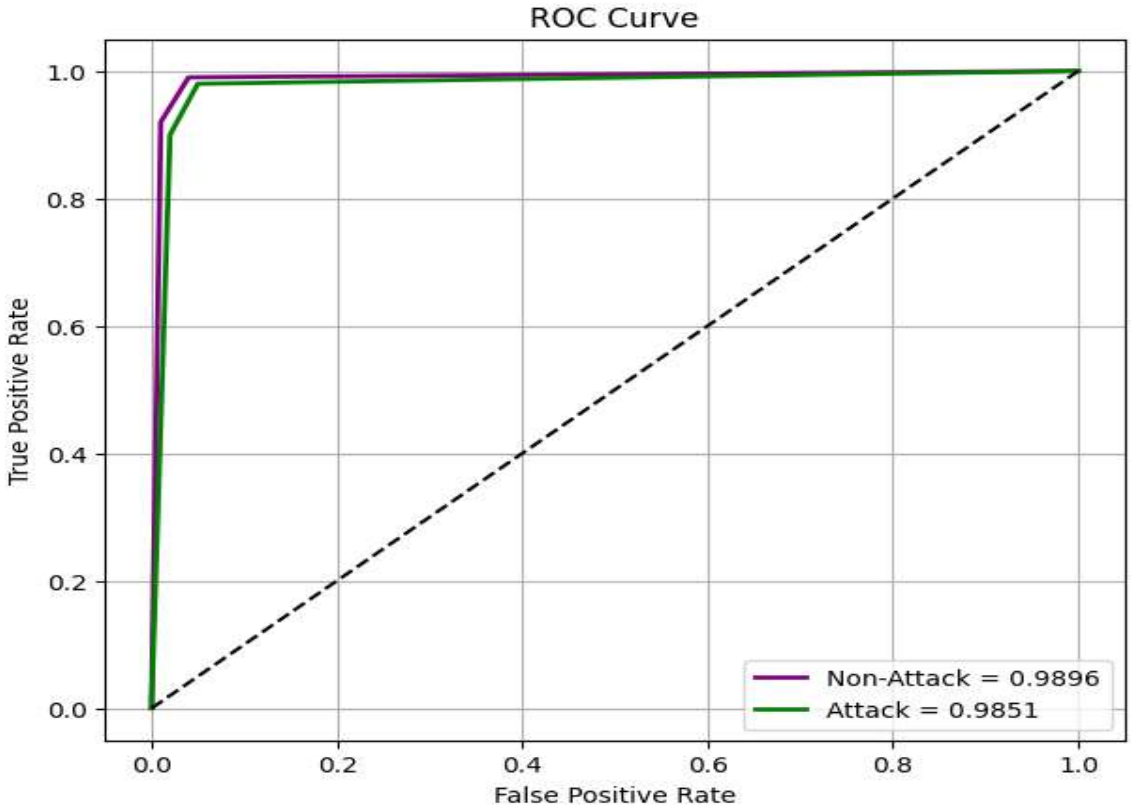


**Figure 9:** ROC Curve of CNN model

Figure 9 displays the ROC curve of the CNN model, which shows how well it can differentiate between the 'Attack' and 'Non-Attack' groups. The purple "non-attack" line and the green "attack" line are positioned around the graph's upper left corner. With a low FPR and a high TPR across many classification criteria, this spot reflects an efficient model. This remarkable performance is confirmed by the AUC values, which are 0.9896 for the non-attack case and 0.9851 for the attack case. The model's ACC in classifying events as positive or negative is strongly shown by an AUC value near to 1.0. The dashed diagonal line represents a random classifier; the significant distance between the model's curves and this baseline further highlights its superior predictive capability. This almost perfect ROC curve indicates that CNN model is very trustworthy in the task of classification.

**Table 3:** comparison between base and propose model Performance matrix for Anomaly Detection Techniques in Network Security

| **Matrix** | **CNN** | **LSTM**[16] | **ET**[3] | **LR**[17] |
|---|---|---|---|---|
| **Accuracy** | **98.73** | 87 | 97.9 | 69.92 |
| **Precision** | **96.61** | 88 | 97.71 | 70.36 |
| **Recall** | **96.44** | 91 | 97.73 | 69.92 |
| **F1-score** | **93.98** | 96 | 97.73 | 69.99 |

Table 3 informs the performance comparison between the performance of anomaly techniques used in network security between the proposed CNN model and other benchmark models of LSTM, ET and LR. The results are definitive since even the CNN model outshines the other alternatives in most of the evaluation measures. Its ACC of (98.73%), has the highest degree of consistency in the detection of the right type of traffic that is normal or an attack. Similarly, its PRE (96.61) and REC (96.44) demonstrate a good tradeoff between false alarm and detection of real cases of attacks. Despite the competitive score of F1 (96%) on LSTM, CNN has a very high rate in the F1-score (93.98%), as well as general higher accuracies and stabilities in guiding values. In the comparison to ET which unlike CNN delivers amazing ACC score of 97.9, and LR which slightly falls behind with the ACC score of only 69.92, the CNN model was proven to be healthy and effective to place itself in the network safety context as an excellent and efficient tool in anomaly detection.

 

### 4.1 Comparative Analytics

The performance of the proposed CNN model and the baseline approaches of LSTM, ET, and LR that were used in this paper clearly give strong reasons to support the fact that the CNN was useful in terms of network security when it comes to detecting anomalies. Although both models, i.e., ET and LSTM, demonstrated competitive ACC results (97.9% and 97.4%, respectively), ET exhibited moderate variability concerning the remaining measures, with LSTM having the best results in terms of REC and F1. As opposed to both of them, CNN demonstrated a balanced performance in terms of all metrics. LR by comparison has considerably lower ACC of 69.92, and thus, is not suitable in a complex task of intrusion detection. The CNN model has a higher ACC of 98.73, PRE of 96.61, and REC of 96.44 with an outstanding ROC-AUC value of 0.98 as compared to the other models which proves the high degree of reliability in the correct classification of an attack or an non-attack traffic. These results not only prove that CNN is more effective when it comes to generalization, but they also provide more accurate trade-off between high sensitivity of detection and low false alarms than the other methods discussed.

## 5 CONCLUSION AND FUTURE SCOPE

Anomaly predicting is a significant aspect of modern cybersecurity, as it allows to anticipate the failure of the system in advance and insure important setups and vulnerable information against possible attacks. In this research, an anomaly detection system that relies on CNN was developed to contribute to the increasing demand in regards to effective security systems. The framework attained balanced learning through the incorporation of pre-processing methods, i.e. SMOTE balancing, feature selection, and normalization and it led to improved detection performance. The CNN proposed had high PRE, REC and F1 of 98.73% in the UNSW-NB15 dataset with improved results compared to the baseline models which included LSTM, Extra Trees and LR. These results show the extent of the model efficiency in intrusion detection and the areas where it can be implemented within the security circles in the real world. However, the flexibility to the previously never realized and dynamic forms of attacks in active settings will need further study. The prospective areas of work are the investigation of hybrid and ensemble approaches to combining CNNs with other deep architecture to become more robust. Scalability may be strengthened by the implementation of the innovative feature engineering and dimensionality reduction methods, particularly concerning big and complicated data. Besides, the implementation of the framework into IoT and cloud architecture will increase its usability. By enhancing interpretability and trust, XAI can be a great fit for large-scale cybersecurity systems that are constantly evolving.

## REFERENCES


[1] N. A. Santoso, R. Lutfayza, B. I. Nughroho, and G. Gunawan, “Anomaly detection in network security systems using machine learning,” *J. Intell. Decis. Support Syst.*, vol. 7, no. 2, pp. 113–120, 2024, doi: 10.35335/idss.v7i2.238.

[2] R. Q. Majumder, “A Review of Anomaly Identification in Finance Frauds Using Machine Learning Systems,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 5, no. 10, pp. 101–110, 2025, doi: 10.48175/IJARSCT-25619.

[3] M. A. Talukder *et al.*, “Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction,” *J. Big Data*, 2024, doi: 10.1186/s40537-024-00886-w.

[4] V. Thangaraju, “Enhancing Web Application Performance and Security Using AI-Driven Anomaly Detection and Optimization Techniques,” *Int. Res. J. Innov. Eng. Technol.*, vol. 09, no. 03, pp. 205–212, 2025, doi: 10.47001/IRJIET/2025.903027.

[5] P. Schummer, A. del Rio, J. Serrano, D. Jimenez, G. Sánchez, and Á. Llorente, “Machine Learning-Based Network Anomaly Detection: Design, Implementation, and Evaluation,” *AI*, vol. 5, no. 4, pp. 2967–2983, 2024, doi: 10.3390/ai5040143.

[6] H. Kali, “The Future Of Hr Cybersecurity: Ai-Enabled Anomaly Detection In Workday Security,” *Int. J. Recent Technol. Sci. Manag.*, vol. 8, no. 6, pp. 80–88, 2023.

[7] S. Sheng and X. Wang, “Network traffic anomaly detection method based on chaotic neural network,” *Alexandria Eng. J.*, vol. 77, pp. 567–579, Aug. 2023, doi: 10.1016/j.aej.2023.07.019.

[8] V. Shah, “Network Verification Through Formal Methods : Current Approaches and Open Issues,” *Int. J. Res. Anal. Rev.*, vol. 8, no. 3, 2021.

[9] N. Roy, R. G. Tiwari, and S. Roy, “Stacking-Based Machine Learning Approach for Anomaly Detection in Embedded System Network Traffic,” in *2025 Fourth International Conference on Smart Technologies, Communication and Robotics (STCR)*, IEEE, May 2025, pp. 1–6. doi: 10.1109/STCR62650.2025.11019443.

[10] S. Ness, V. Eswarakrishnan, H. Sridharan, V. Shinde, N. V. P. Janapareddy, and V. Dhanawat, “Anomaly Detection in Network Traffic using Advanced Machine Learning Techniques,” *IEEE Access*, vol. 13, no. December 2024, pp. 16133–16149, 2025, doi: 10.1109/ACCESS.2025.3526988.

[11] A. Sharma, R. Bajaj, and R. Sahu, “Intelligent Approach for Anomaly Detection using Machine learning Techniques in Industrial Control system,” in *2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI)*, IEEE, Nov. 2024, pp. 1–6. doi: 10.1109/IDICAIEI61867.2024.10842786.

[12] V. N and P. T. V Bhuavneswari, “ADBIS: Anomaly Detection to Bolster IoT Security Using Machine Learning,” in *2023 IEEE 3rd International Conference on Applied Electromagnetics, Signal Processing, & Communication (AESPC)*, Nov. 2023, pp. 1–6. doi: 10.1109/AESPC59761.2023.10390100.

[13] A. K. Gulhare, A. Badholia, and A. Sharma, “Mean-Shift and Local Outlier Factor-Based Ensemble Machine Learning Approach for Anomaly Detection in IoT Devices,” in *2022 International Conference on Inventive Computation*


 

*Technologies (ICICT)*, IEEE, Jul. 2022, pp. 649–656. doi: 10.1109/ICICT54344.2022.9850880.

[14] R. Singh and G. Srivastav, "Novel Framework for Anomaly Detection Using Machine Learning Technique on CIC-IDS2017 Dataset," in *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, IEEE, Nov. 2021, pp. 632–636. doi: 10.1109/ICTAI53825.2021.9673238.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, 2017, doi: 10.1145/3065386.

[16] R. Almarshdi, L. Nassef, E. Fadel, and N. Alowidi, "Hybrid Deep Learning Based Attack Detection for Imbalanced Data Classification," *Intell. Autom. Soft Comput.*, vol. 35, no. 1, pp. 297–320, 2023, doi: 10.32604/iasc.2023.026799.

[17] S. A. Ajagbe, J. B. Awotunde, and H. Florez, "Intrusion Detection: A Comparison Study of Machine Learning Models Using Unbalanced Dataset," *SN Comput. Sci.*, vol. 5, no. 8, p. 1028, Nov. 2024, doi: 10.1007/s42979-024-03369-0.

 