# Traffic-Aware Adaptive Intrusion Detection System for Dynamic Network Conditions

Satish Dekka[1], P. Pavani[2], S. Kamal Kumar[3], V. Jahnavi[4], S. Sirisha[5]
[1]*Department of Computer Science and Engineering Lendi Institute of Engineering and Technology Vizianagaram, India*
*doi.org/10.64643/IJIRTV12I12-202562-459*

***Abstract*—The Intrusion Detection System is an indispensable component in ensuring security in computer networks. At present, due to the rapidly growing complexity of networks, it becomes challenging to detect any kind of intrusion accurately. Most intrusion detection systems rely on using a single algorithmic approach to differentiate between normal and suspicious network activities. Despite achieving satisfactory results under some conditions, such an approach loses its effectiveness in different situations caused by varying network traffic conditions. Cases of predominant normal traffic, sudden attacks, or complex network conditions can reduce the accuracy of such a method and raise the number of false positives. Therefore, this paper suggests an adaptive approach for intrusion detection systems to deal with dynamic conditions in computer networks. It involves using multiple machine learning and deep learning approaches, including decision trees, random forests, convolutional neural networks (CNNs), long short-term memory (LSTM), and hybrid CNN-LSTMs. Common intrusion detection datasets will be used for analysis in order to measure the effectiveness of each methodical approach and replicate various network traffic conditions. Then, based on the achieved accuracy, precision, recall, and false positive results, the optimal solution will be chosen. The experimental results demonstrate that the adaptive method enhances detection accuracy, reduces the false alarm rate, and fortifies the overall robustness of the system in comparison to traditional static intrusion detection systems. This problem can be solved well by using the adaptive model.**

***Index Terms*—Intrusion Detection Systems, Network Security, Adaptive Detection, Machine Learning, Deep Learning, Convolutional Neural Networks, Long Short-Term Memory**

## I. INTRODUCTION

Technology advancements and networking applications have made significant progress in the field of computer networks. Computer networks help make a lot of useful services possible, such as online banking, cloud computing, e-commerce, communication, and social media. This change has made it easy for people all over the world to talk to each other and get information from anywhere in the world. But as more people use computer networks, new problems arise, especially when it comes to keeping the networks safe.

A lot of cybercriminals keep trying to use the flaws in networking systems to attack computer networks. There are many kinds of network attacks, such as denial-of-service attacks, probing attacks, remote-to-local attacks, and user-to-root attacks. This could lead to a lot of big problems and losses, including a data breach and the network's service being interrupted. So, it's important to put some measures in place to keep an eye on network traffic so that you can spot any suspicious activity right away.

The intrusion detection systems are vital in ensuring net-work security through the analysis of the network traffic and identification of any possible attacks. Intrusion detection system techniques are usually categorised as signature-based detections for known attacks and anomaly-based detections for abnormal behaviour [19]. Traditional intrusion detection sys-tems, although popularly implemented in many organisations, face numerous challenges. Standard IDS models use set rules and algorithms that can't change to deal with new or unknown attacks [7].

Traffic conditions in a network tend to change over time. Some periods of the day might be characterised by normal traffic, while other moments may include high malicious traffic. It would, therefore, be difficult for one model to provide accurate and effective results for all the different situations.

To address this challenge, there have been efforts to use artificial intelligence, machine learning, and

deep learning methods in developing intrusion detection systems. Algorithms that are commonly used in this case include decision tree and random forest approaches [19]. Also, deep learning architectures like CNN and LSTM have shown that they can recognise complex spatial-temporal patterns in network traffic patterns very well [7], [10]. Combining the strengths of both deep learning and traditional methods, such as CNN-LSTM, can improve detection even more [4], [5].

However, despite the progress, the existing solutions con-tinue to use one type of algorithmic model for traffic classifica-tion. Such an approach may not always ensure optimal results for different types of network traffic data. As a solution to this problem, this research introduces a novel traffic adaptive IDS that can operate efficiently in highly dynamic network environments.

Rather than following one particular model of machine learning or deep learning, the proposed solution makes an evaluation of several models and selects the most appropriate model according to traffic data. Models tested in this work include Decision Tree, Random Forest, CNN, LSTM, and a hybrid CNN-LSTM. The framework starts with data pre-processing of the selected dataset. For data training, NSL-KDD dataset is chosen as one of the most commonly used benchmarks [10]. After each model undergoes training, it will be tested using performance metrics like accuracy, precision, recall, and false positive rate. Given the nature of the traffic, the adaptive model selection process will pick out the best model for detecting intrusions. The benefit of this approach is that it increases the efficiency of the intrusion detection system.

### *A. Research Gap*

Although numerous intrusion detection systems have been proposed using machine learning and deep learning tech-niques, a majority of existing approaches depend on a single model for traffic classification. While such models may deliver high performance under specific or controlled conditions, they often fail to maintain consistent accuracy when network traffic patterns vary over time. In real-world environments, network traffic is highly dynamic due to variations in user behavior, system workload, and the presence of cyber attacks. As a result, a static detection model may not be capable of handling all traffic scenarios effectively. Therefore, there is a need for an advanced intrusion detection framework that can adapt to changing network conditions and dynamically choose the most appropriate detection model for different traffic situations.

### *B. Objectives of the Study*

The main objectives of this study are:

- To design and implement an efficient intrusion detection system capable of distinguishing between normal and malicious network traffic.
- To develop and assess multiple machine learning and deep learning models for intrusion detection tasks.
- To examine network traffic behavior under various simulated conditions.
- To measure the performance of different models using evaluation metrics such as accuracy, precision, recall, and false positive rate.
- To design an adaptive mechanism that dynamically selects the most appropriate model based on changing network traffic conditions.

### *C. Main Contributions*

1. Development of a Traffic-Aware Adaptive Intrusion Detection System designed to handle varying network traffic conditions effectively. 2. Integration of multiple machine learning and deep learning techniques, including Decision Tree, Random Forest, CNN, LSTM, and a hybrid CNN-LSTM model for intrusion detection. 3. Design and implementation of an adaptive model selection mechanism that dynamically chooses the most suitable detection model based on current network traffic characteristics. 4. Evaluation of the proposed system using the NSL-KDD dataset along with performance metrics such as accuracy, precision, recall, and false positive rate. 5. Enhancement of intrusion detection performance by minimizing false alarms and improving overall system reliability in dynamic network environments.

### *D. Organization of the Paper*

The structure of the paper is organized as follows. Section
II provides the literature review, covering existing studies related to intrusion detection systems and machine learning-based security techniques. Section III explains the proposed methodology of the

Traffic-Aware Adaptive Intrusion Detection System, including system architecture, dataset details, data preprocessing steps, intrusion detection models, adaptive model selection approach, and experimental setup. Section IV presents the results and discussion, where the performance of various machine learning and deep learning models is evaluated and compared using the NSL-KDD dataset. Section V concludes the paper by summarizing the major findings and contributions of the proposed system. Section VI discusses potential future enhancements and possible extensions of the proposed framework. Finally, Section VII lists the references used in this study.

## II. LITERATURE REVIEW

Network intrusion detection has become an important re-search area due to the rapid increase in cyber threats and the growing complexity of network environments. Researchers have explored various machine learning and deep learning techniques to improve the detection accuracy and adaptability of intrusion detection systems. Recent studies have focused on developing intelligent IDS models capable of handling dynamic network traffic, unknown attacks, and large scale datasets. Several studies have investigated the use of ad-vanced deep learning architectures for intrusion detection. Liu proposed a dynamic network intrusion detection model that combines a Transformer architecture with an Adversar-ial Autoencoder(AAE) to enhance feature representation and improve attack detection. The Transformer model captures long term dependencies in network traffic using a self at-tention mechanism, while the adversarial autoencoder reduces overfitting and improves detection of unknown attacks. Ex-perimental evaluation on KDD99 and CICIDS2017 datasets showed that the proposed model achieves higher detection accuracy and lower false positive rates compared to traditional IDS methods [1]. Similarly, Villegas Ch et al. developed an adaptive deep learning based intrusion detection system designed to improve threat detection in dynamic cyberse-curity environments. Their system continuously learns from network traffic data and adapts to evolving cyber threats. Experimental results demonstrated significant improvements in detection accuracy, precision, recall, and response time when compared to traditional rule based intrusion detection approaches [2]. Machine learning techniques have also been widely applied in intrusion detection systems. Thaseen and Kumar developed a machine learning based IDS using algo-rithms such as Na¨ıve Bayes, Decision Tree, and Support Vector Machine for detecting malicious network activities. Their experimental analysis showed that machine learning classifiers can effectively improve intrusion detection performance [19]. Cantone et al. conducted a cross dataset generalization study to analyze the robustness of machine learning based intrusion detection models. Their research evaluated multiple classifiers across different intrusion detection datasets and found that although models perform well on the dataset used for training, their performance decreases significantly when applied to unseen datasets. The study emphasized the importance of designing IDS models capable of generalizing across different network environments [13], [18]. Deep learning models have demonstrated promising performance in identifying complex network attack patterns. Du et al. proposed a CNN LSTM based intrusion detection model known as NIDS CNNLSTM for Industrial Internet of Things environments. The model combines convolutional neural networks for spatial feature extraction with long short term memory networks to capture temporal traffic patterns. Experimental results on KDD Cup 99, NSL KDD, and UNSW NB15 datasets showed improved classification accuracy and reduced false alarm rates compared to conventional IDS methods [4], [11]. Hybrid deep learning models have also been explored to enhance detection perfor-mance. Altunay and Albayrak developed a hybrid CNN LSTM intrusion detection system for industrial IoT networks. Their model extracts spatial and temporal features from network traffic and achieves higher accuracy and lower loss compared to individual CNN and LSTM models [5]. Abdulmajeed and Husien introduced MLIDS22, a hybrid CNN LSTM intrusion detection system trained using mixed datasets. By combining CIC IDS2017 and CSE CIC IDS2018 datasets, their approach improves model generalization and robustness against diverse cyber threats [6]. Wang conducted a comprehensive study on deep learning based intrusion detection techniques, focusing on CNN, LSTM, and hybrid CNN LSTM architectures. The study highlighted that hybrid models outperform traditional machine learning algorithms in terms of detection accuracy and false positive reduction [7]. In the context of

IoT envi-ronments, Sinha et al. proposed a high performance hybrid LSTM CNN architecture to improve intrusion detection accu-racy. The system was evaluated on the BoT IoT dataset and achieved an accuracy of 99.87Chatterjee et al. investigated the effectiveness of deep learning techniques including artificial neural networks, convolutional neural networks, and long short term memory networks for network intrusion detection. Their experiments showed that deep learning models significantly improve detection performance by capturing complex traffic patterns [9]. Earlier research by Vinayakumar et al. explored deep neural networks for intelligent intrusion detection. Their study evaluated deep learning models on multiple benchmark datasets including NSL KDD, UNSW NB15, and CICIDS2017 and demonstrated that deep neural networks outperform tra-ditional machine learning models in detecting sophisticated cyber attacks [10]. Adaptive intrusion detection mechanisms have also been proposed to address challenges associated with dynamic network traffic. Almania et al. developed an adaptive IDS that integrates KNN classification, fuzzy clustering, and ensemble classifiers such as Decision Tree and Random Forest. The system dynamically triggers retraining based on traffic variations and achieved high detection accuracy with reduced false alarm rates [11]. Belarbi et al. proposed a federated deep learning approach for intrusion detection in IoT networks to address privacy and scalability issues of centralized IDS architectures. Their model uses distributed learning across multiple devices and aggregates model updates using federated learning techniques [12]. Security researchers have also inves-tigated methods to improve the robustness of deep learning based IDS models. Yuan et al. proposed a framework that integrates adversarial attack detection techniques with deep learning intrusion detection systems to improve IDS reliability against adversarial attacks [13]. Dynamic intrusion detection systems capable of identifying unknown attacks have also been explored. Xing et al. developed a dynamic intrusion detection system using Transformer and CNN architectures to extract global and local traffic features and detect previously unseen attacks [14]. In cloud computing environments, Jabez et al. proposed an adaptive intrusion detection mechanism that combines convolutional neural networks with support vector machines to handle large scale traffic data and improve detection accuracy [15]. Aliyu et al. introduced a decentralized and self adaptive intrusion detection framework that integrates deep neural networks, blockchain technology, and continuous learning mechanisms to enhance IDS reliability and security [16]. Harish and Annapurna proposed an enhanced intrusion detection system that integrates ADASYN oversampling with residual neural network architectures to address the issue of imbalanced datasets and improve detection performance [17]. Recent survey studies have also analyzed the role of machine learning and deep learning techniques in intrusion detection systems. Cherukuri et al. reviewed several IDS approaches and concluded that hybrid and deep learning models generally achieve higher accuracy and lower false positive rates com-pared to traditional intrusion detection techniques [20]

### *A. Summary of Literature Survey*

Furthermore, the comparison of existing intrusion detection approaches reveals that no single model can consistently perform well under all network traffic conditions. Traditional machine learning models offer faster computation and sim-plicity, whereas deep learning models provide better capability in capturing complex patterns within network data. However, each approach has its own limitations in terms of scalability, computational cost, and adaptability to dynamic environments. In addition, recent research has focused on hybrid and adaptive intrusion detection systems that combine multiple techniques to improve detection performance. These systems aim to dynamically adjust to changing network behaviors and provide more reliable security solutions. By analyzing these studies, it becomes evident that an adaptive framework capable of selecting appropriate models based on traffic conditions can significantly enhance intrusion detection accuracy and reduce false alarm rates.

TABLE I: SUMMARY OF LITERATURE SURVEY

| No | Author(s) | Year | Model | Dataset | Key Contribution | Limitation |
|---|---|---|---|---|---|---|
| 1 | Weiwei Liu | 2024 | Transformer + Adversarial Autoencoder | KDD99, CICIDS2017 | Improved detection accuracy using self-attention and feature optimiza-tion. | High computational complexity for real-time systems. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2 | Villegas-Ch et al. | 2024 | Adaptive Deep Learning IDS | Standard IDS datasets | Adaptive IDS capable of learning evolving cyber threats. | Requires continuous training and high computational resources. |
| 3 | Shahid et al. | 2024 | Hybrid ML/DL IDS | ROUT-4-2023 | Achieved high accuracy for IoT network attack detection. | Focused mainly on IoT routing at-tacks. |
| 4 | Du et al. | 2023 | CNN-LSTM | KDD99,NSL-KDD, UNSW-NB15 | Extracts spatial and temporal fea-tures for IIoT intrusion detection. | High training time and computa-tional overhead. |
| 5 | Altunay & Albayrak | 2023 | CNN + LSTM Hybrid | UNSW-NB15, X-IIoTID | Improved detection accuracy in In-dustrial IoT environments. | Model complexity increases sys-tem cost. |
| 6 | Abdulmajeed & Husien | 2022 | CNN-LSTM Hybrid | CIC-IDS2017, CSE-CIC-IDS2018 | Mixed datasets improve IDS ro-bustness. | Dataset mixing increases training complexity. |
| 7 | Wang | 2024 | CNN, LSTM, CNN-LSTM | KDD99, CICIDS 2017, UNSW-NB15 | Comparative study of deep learning IDS models. | Limited focus on adaptive IDS strategies. |
| 8 | Sinha et al. | 2025 | LSTM-CNN Hybrid | BoT-IoT | Achieved very high accuracy for IoT intrusion detection. | Designed specifically for IoT net-works. |
| 9 | Chatterjee et al. | 2024 | ANN, CNN, LSTM | CICIDS2017 | Demonstrated deep learning effec-tiveness in IDS. | Performance depends heavily on dataset quality. |
| 10 | Vinayakumar et al. | 2019 | Deep Neural Network | Multiple datasets | Scalable deep learning IDS frame-work. | High computational requirements for deployment. |
| 11 | Almania et al. | 2025 | Ensemble ML (KNN, DT, RF) | Benchmark datasets | Adaptive IDS with retraining mechanism. | Limited evaluation on large-scale real-time traffic. |
| 12 | Belarbi et al. | 2023 | Federated Deep Learning | TON-IoT | Privacy-preserving distributed IDS architecture. | Federated training may increase communication overhead. |
| 13 | Yuan et al. | 2023 | DL + Adversarial Detec-tion | NSL-KDD, CICIDS2018 | Improved robustness against adver-sarial attacks. | Detection complexity increases system latency. |
| 14 | Xing et al. | 2023 | Transformer + CNN | UNSW-NB15 | Detects unknown attacks using dy-namic learning. | Requires large training datasets. |
| 15 | Jabez et al. | 2024 | CNN + SVM | Cloud datasets | Adaptive IDS for cloud-hosted big data systems. | Limited evaluation on heteroge-neous network environments. |
| 16 | Aliyu et al. | 2024 | Blockchain + Deep Learn-ing | NSL-KDD | Decentralized IDS with secure in-trusion data storage. | Blockchain integration increases computational cost. |
| 17 | Harish & Annapurna | 2025 | ResNet + ADASYN | UNSW-NB15 | Handles imbalanced datasets effec-tively. | Slightly lower accuracy compared to hybrid DL models. |
| 18 | Cantone et al. | 2024 | ML Cross-Dataset Study | CIC, LycoS datasets | Demonstrated generalization issues across datasets. | Does not propose a new IDS model. |

| | | | | | | |
|---|---|---|---|---|---|---|
| 19 | Jadhav et al. | 2024 | ML (NB, DT, KNN) | KDD Cup 99 | Demonstrates effectiveness of clas-sical ML models. | Lower performance compared to deep learning models. |
| 20 | Almohaimeed et al. | 2025 | ML & DL Survey | Multiple datasets | Comprehensive review of IDS techniques for IoT. | Does not implement a practical IDS model. |

Table I provides a consolidated overview of key research contributions in the field of intrusion detection systems. It highlights the different models used, datasets considered, major contributions, and associated limitations of each study, enabling a clear comparison of existing approaches. This comparison helps in identifying research trends and under-standing the strengths and weaknesses of various techniques. It also provides a foundation for developing more effective and adaptive intrusion detection frameworks.

## III. PROPOSED METHODOLOGY

This work presents a Traffic Aware Adaptive Intrusion Detection System aimed at enhancing detection performance under dynamic network conditions. Conventional intrusion detection systems generally depend on a single classification model, which may not maintain consistent performance when traffic patterns vary over time. To address this issue, the proposed approach utilizes multiple machine learning and deep learning models and selects the most suitable one dynamically based on current traffic characteristics.

The overall framework is composed of several key stages, including dataset preparation, preprocessing, model training, traffic condition evaluation, adaptive model selection, and final intrusion detection. The system continuously analyzes incoming network traffic and applies the most appropriate model to classify activities as normal or malicious. This adaptive mechanism helps improve detection accuracy and ensures better performance across varying network scenarios.

### *A. System Overview*

The proposed intrusion detection framework is composed of multiple stages, including network traffic input, data prepro-cessing, feature extraction, model training, model evaluation, traffic condition analysis, adaptive model selection, intrusion detection, and alert generation.

At the initial stage, the system obtains network traffic data from the NSL KDD dataset, which includes labeled instances of both normal traffic and various attack types. The dataset undergoes preprocessing to eliminate inconsistencies and convert it into a suitable format for machine learning models. Following preprocessing, relevant features are extracted to support effective model training.

Once the models are trained, their performance is assessed using evaluation metrics such as accuracy and confusion matrix. Based on the analysis of current traffic conditions, the system dynamically selects the most appropriate model for intrusion detection. Finally, the selected model is used to classify incoming traffic, and alerts are generated whenever malicious activities are identified.

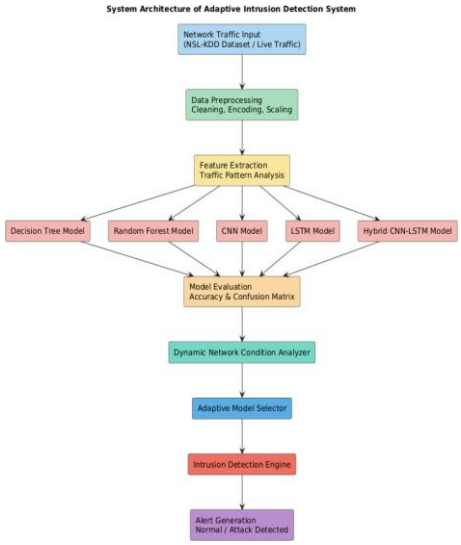


Fig. 1. Architecture of the Traffic-Aware Adaptive Intrusion Detection System

### *B. Dataset Description*

The proposed system utilizes the NSL KDD dataset, which is a commonly used benchmark dataset for evaluating intrusion detection systems. It is an enhanced version of the original KDD Cup 1999 dataset, designed to overcome issues such as redundant entries and biased data distribution. Each record in the NSL KDD dataset consists of 41 features that represent different properties of network connections. These features include basic attributes, content-based information, and sta-tistical measures related to network behavior. The dataset contains both normal traffic instances and multiple categories of attacks, including: 1. Denial of Service (DoS) 2. Probe attacks 3. Remote to Local (R2L) attacks 4. User to Root (U2R) attacks The use of this dataset enables the proposed system

to effectively evaluate model performance across a variety of attack types and network traffic conditions.

*C. Data Preprocessing*

Data preprocessing plays a crucial role in preparing the dataset for machine learning and deep learning models. The NSL KDD dataset contains both numerical and categorical attributes, which require multiple preprocessing steps before model training. Initially, data cleaning is carried out to eliminate duplicate, inconsistent, or irrelevant records. This process enhances data quality and reduces noise in the dataset. Next, label encoding is applied to transform categorical features such as protocol type, service, and flag into numerical values, as most machine learning algorithms require numerical input. Subsequently, feature scaling and normalization techniques are applied to ensure that all features are within a similar range.

This step prevents features with larger values from dominating the learning process and improves model performance. Finally, the dataset is split into training and testing sets using a train-test split method. The training set is used to build the models, while the testing set is used to evaluate their effectiveness.

*D. Feature Extraction and Traffic Pattern Analysis*

Following the preprocessing stage, significant features are extracted from the dataset to facilitate traffic pattern analysis. The main objective of feature extraction is to select the most relevant attributes that effectively distinguish between normal and malicious network activities. Subsequently, traffic pattern analysis is performed to gain insights into the behavior of net-work data. By studying these patterns, the system can identify unusual or abnormal activities that may indicate potential security threats. This step enhances the overall performance of the models by supplying meaningful and well-structured features, thereby improving the accuracy of intrusion detection.

*E. Intrusion Detection Models*

The proposed system incorporates multiple machine learning and deep learning models for identifying malicious activities in network traffic. These models process extracted traffic features and perform classification to distinguish between normal and abnormal behavior.

The models considered in this work include Decision Tree, Random Forest, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and a Hybrid CNN-LSTM model. Each of these models contributes uniquely to intrusion detection by capturing different patterns and characteristics present in network data.

*1) Decision Tree (DT)*

Decision Tree is a supervised learning algorithm used for classification by constructing a hierarchical tree structure based on decision rules. It is simple to understand and enables fast classification, making it suitable for analyzing network traffic data.

Entropy:

$$H(S) = -\sum_{i=1} p_i \log_2(p_i) \quad (1)$$

where $p_i$ denotes the probability of class $i$.

Information Gain:

$$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v) \quad (2)$$

where $S$ represents the dataset, $A$ indicates the selected attribute, and $S_v$ refers to the subset of data corresponding to value $v$.

*2) Random Forest (RF)*

Random Forest is an ensemble-based machine learning tech-nique that builds multiple decision trees and combines their outputs to enhance classification performance and minimize overfitting. Due to its robustness and ability to handle high-dimensional data, it is widely applied in intrusion detection tasks.

The final prediction of the Random Forest model is deter-mined through a majority voting mechanism:

$$\hat{y} = \text{mode}(h_1(x), h_2(x), \ldots, h_n(x)) \quad (3)$$

where $h_i(x)$ denotes the prediction of the $i^{th}$ decision tree, and $n$ represents the total number of trees in the ensemble.

*3) Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) is a deep learning model designed to automatically extract significant features from input data. In intrusion detection systems, CNN is particularly effective in capturing spatial relationships within network traffic features.

Convolution Operation:

$$y(i, j) = \sum_m \sum_n x(i + m, j + n) \cdot w(m, n) \quad (4)$$

where $x$ represents the input feature map, $w$ denotes the convolution kernel (filter), and $y$ is the resulting output feature map.

Activation Function (ReLU):

$$f(x) = \max(0, x) \quad (5)$$

*4) Long Short-Term Memory (LSTM)*

Long Short-Term Memory (LSTM) is a type of recurrent neural network that is designed to learn sequential dependen-cies in time-series data. It is particularly useful in intrusion detection for analyzing ordered network traffic and identifying temporal attack patterns.

Forget Gate:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (6)$$

Input Gate:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (7)$$

Cell State Update:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (8)$$

where $f_t$ denotes the forget gate, $i_t$ represents the input gate, and $C_t$ indicates the updated cell state.

*5) Hybrid CNN-LSTM Model*

The Hybrid CNN-LSTM model integrates the feature ex-traction strength of CNN with the sequence learning capability of LSTM. This combined architecture allows the system to capture both spatial and temporal characteristics of network traffic, enabling more accurate detection of complex intrusion patterns compared to individual models.

CNN extracts spatial features:

$$F_{spatial} = CNN(X) \quad (9)$$

LSTM captures temporal dependencies:

$$F_{temporal} = LSTM(F_{spatial}) \quad (10)$$

Final Classification:

$$y = Softmax(W \cdot F_{temporal} + b) \quad (11)$$

*F. Model Evaluation*

After completing the training phase, the effectiveness of each intrusion detection model is assessed using the testing dataset. Performance evaluation is carried out using metrics such as accuracy and the confusion matrix. These measures provide insight into how well the models are able to correctly classify network traffic as either normal or malicious.

*G. Dynamic Network Condition Analysis and Adaptive Model Selection*

A major contribution of the proposed system is the adaptive model selection mechanism. Unlike traditional approaches that depend on a single detection model, this system evaluates mul-tiple trained models under varying network traffic conditions. The system examines the characteristics of incoming traffic to identify the current scenario, such as normal-dominant, attack-dominant, or mixed traffic conditions. Based on this analysis, the adaptive module selects the model that is most suitable for the identified traffic pattern. This dynamic selection strategy helps the intrusion detection system maintain consistent accuracy while minimizing false alarms, making it more effective in handling changing network environments.

*H. Intrusion Detection and Alert Generation*

After the adaptive module selects the most appropriate model, it is applied to classify the incoming network traffic. The system processes the traffic data and determines whether it represents normal behavior or a potential intrusion. When any suspicious or malicious activity is identified, the system triggers an intrusion alert and informs the network adminis-trator. This alerting mechanism enables timely response and allows necessary actions to be taken to safeguard the network from potential threats.

*I. Experimental Setup*

The experiments were carried out using the Python pro-gramming language along with popular machine learning and deep learning libraries such as Scikit-learn, TensorFlow, and Keras. The proposed models were trained and tested using the NSL-KDD dataset, which was divided into training and testing sets in a 70:30 ratio. Machine learning algorithms, in-cluding Decision Tree and Random Forest, were implemented using the Scikit-learn library. Deep learning models such as CNN, LSTM, and the Hybrid CNN-LSTM were developed using TensorFlow and Keras frameworks. The performance of all models was assessed using evaluation metrics including accuracy, precision, recall, F1-score, and confusion matrix, providing a comprehensive analysis of their classification effectiveness.

*J. Algorithms*

This section describes the algorithms implemented in the proposed intrusion detection system. Each algorithm outlines the steps involved in training and testing, which are used to classify network traffic as either normal or malicious.

Algorithm 1 Decision Tree Based Intrusion Detection

1: Input: Preprocessed dataset $X$, labels $Y$
2: Output: Predicted class labels
3: Load the NSL-KDD dataset
4: Apply preprocessing techniques

5: Divide the dataset into training and testing subsets
6: Initialize the Decision Tree classifier
7: Train the model using the training dataset
8: for each test instance $x_i$ do
9: Evaluate the instance using decision rules
10: Assign the corresponding class label (Normal / Attack)
11: end for
12: Calculate performance metrics such as Accuracy and Confusion Matrix
13: Return predicted class labels

Algorithm 2 Random Forest Based Intrusion Detection

1: Input: Training dataset, Testing dataset
2: Output: Predicted class labels
3: Load NSL-KDD dataset
4: Perform preprocessing and scaling
5: Split dataset into training and testing sets
6: Initialize Random Forest with $N$ trees
7: for each tree $T_i$ do
8: Select random bootstrap sample
9: Select random feature subset
10: Train decision tree
11: end for
12: for each test sample $x_j$ do
13: Collect predictions from all trees
14: Apply majority voting 15: Assign final class label 16: end for
17: Compute Accuracy and Confusion Matrix

Algorithm 3 CNN Based Intrusion Detection

1: Input: Preprocessed dataset $X$
2: Output: Binary classification (Normal / Attack)
3: Load NSL-KDD dataset
4: Perform preprocessing and normalization
5: Encode categorical attributes
6: Split dataset into training and testing sets
7: Reshape data for CNN input 8: Construct CNN architecture 9: Input layer
10: Convolution layer
11: Max pooling layer
12: Fully connected layer
13: Train CNN model
14: for each test sample do
15: Extract spatial features
16: Classify using sigmoid activation
17: end for
18: Output prediction results

Algorithm 4 LSTM Based Intrusion Detection

1: Input: Preprocessed dataset
2: Output: Detection result 3: Load NSL-KDD dataset 4: Perform preprocessing
5: Normalize input features
6: Split dataset into training and testing sets
7: Reshape dataset into sequential format
8: Construct LSTM architecture
9: Input layer
10: LSTM layer
11: Dense output layer
12: Train LSTM model
13: for each sequence do
14: Capture temporal patterns
15: Predict class label
16: end for
17: Evaluate using Accuracy and Confusion Matrix

Algorithm 5 Hybrid CNN-LSTM Based Intrusion Detection

1: Input: Preprocessed dataset
2: Output: Predicted class labels
3: Load dataset
4: Perform preprocessing and scaling
5: Split dataset into training and testing sets
6: Reshape data for hybrid model
7: Construct Hybrid CNN-LSTM architecture
8: CNN layers for spatial feature extraction
9: LSTM layer for temporal learning 10: Fully connected classification layer 11: Train hybrid model
12: for each test sample do
13: Extract spatial features using CNN
14: Capture temporal dependencies using LSTM
15: Predict traffic class
16: end for
17: Evaluate using Accuracy and Confusion Matrix

Algorithm 6 Adaptive IDS Model Selection

1: Input: Network traffic condition $C$
2: Output: Selected IDS model
3: Monitor incoming network traffic
4: Analyze traffic characteristics
5: Determine condition $C$
6: if $C$ = Normal then

7: Select Decision Tree model
8: else if $C$ = Mixed then
9: Select Random Forest model
10: else
11: Select Hybrid CNN-LSTM model
12: end if
13: Apply selected model for intrusion detection
14: Classify traffic as Normal or Attack
15: Generate alert if attack detected

## IV. RESULTS AND DISCUSSIONS

This section describes the results obtained from implementing the proposed Traffic-Aware Adaptive Intrusion Detection System (IDS). The performance of various machine learning and deep learning models is analyzed using the NSL-KDD dataset. The models evaluated in this study include Decision Tree, Random Forest, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and the Hybrid CNN-LSTM model.
To evaluate the effectiveness of these models, standard performance metrics such as Accuracy, Precision, Recall, and F1-Score are utilized.

### *A. Model Evaluation Metrics*

The performance of the intrusion detection models is evalu-ated using a confusion matrix consisting of four components:

- True Positive (TP): Attack traffic correctly classified as attack
- True Negative (TN): Normal traffic correctly classified as normal
- False Positive (FP): Normal traffic incorrectly classified as attack
- False Negative (FN): Attack traffic incorrectly classified as normal

The evaluation metrics are calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \quad (15)$$

### *B. Decision Tree Model Results*

The Decision Tree model was evaluated using the NSL KDD dataset to classify network traffic into normal and attack categories. The model achieved an accuracy of 99.70%, indicating strong performance in detecting malicious network activities.

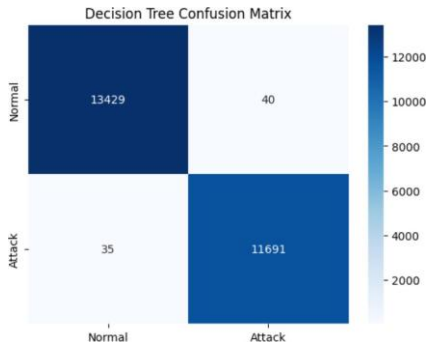


Fig. 2. Performance Visualization of Decision Tree Model

The precision, recall, and F1 score values were approxi-mately 1.00 for both classes, demonstrating that the model effectively identifies intrusion events with minimal misclassi-fication.

TABLE II: CONFUSION MATRIX OF DECISION TREE MODEL

| Actual / Predicted | Normal | Attack |
|---|---|---|
| Normal | 13427 | 42 |
| Attack | 33 | 11693 |

From the confusion matrix, it can be observed that 13,427 normal traffic instances and 11,693 attack instances were correctly classified. Only a small number of samples were misclassified, where 42 normal instances were incorrectly classified as attacks and 33 attack instances were incorrectly classified as normal. These results indicate that the Decision Tree model achieves high detection accuracy with very low false positives and false negatives, making it an effective baseline model for network intrusion detection.

### *C. Random Forest Model Results*

The Random Forest model was implemented to improve intrusion detection performance by utilizing an ensemble learning approach. Random Forest combines multiple decision trees and performs classification using majority voting, which helps reduce overfitting and improves model generalization.

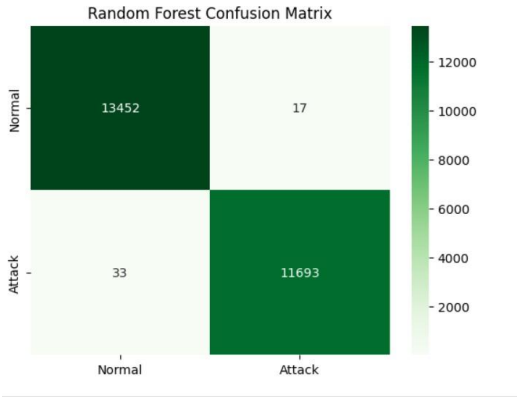


Fig. 3. Performance Visualization of Random Forest Model

The Random Forest model achieved an accuracy of 99.80% on the NSL KDD testing dataset, demonstrating slightly im-proved performance compared to the Decision Tree model. The high accuracy indicates that the model effectively distin-guishes between normal and malicious network traffic.

TABLE III: CONFUSION MATRIX OF RANDOM FOREST MODEL

| Actual / Predicted | Normal | Attack |
|---|---|---|
| Normal | 13452 | 17 |
| Attack | 33 | 11693 |

From the confusion matrix, it can be observed that 13,452 normal traffic instances and 11,693 attack instances were correctly classified. Only 17 normal instances were incorrectly classified as attacks, while 33 attack instances were misclassified as normal traffic. These results indicate that the Random Forest model achieves very high detection accuracy with minimal misclassification, outperforming the Decision Tree model by reducing the number of false positives. The ensemble nature of Random Forest allows it to handle high dimensional network traffic data more effectively, making it highly suitable for intrusion detection tasks.

*D. CNN Model Results*

The Convolutional Neural Network (CNN) model was im-plemented to capture spatial relationships among network traf-fic features for intrusion detection. CNN models are capable of automatically extracting complex feature patterns from high dimensional data through convolution and pooling operations.

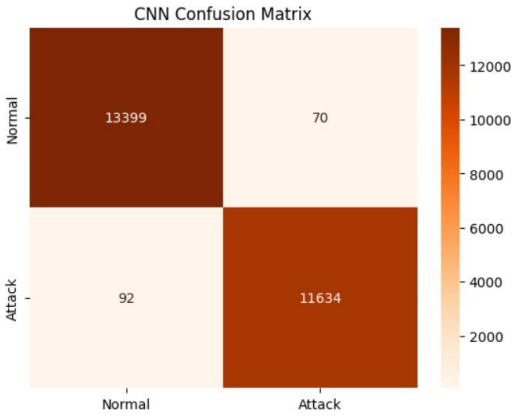


Fig. 4. Confusion Matrix of CNN Model

During training, the CNN model was trained for 10 epochs using the preprocessed NSL KDD dataset. The training process gradually improved the classification performance, with the training accuracy increasing from 97.23% in the first epoch to approximately 99.36% in the final epoch. Similarly, the validation accuracy increased steadily, reaching approximately 99.46%, indicating good generalization of the model.

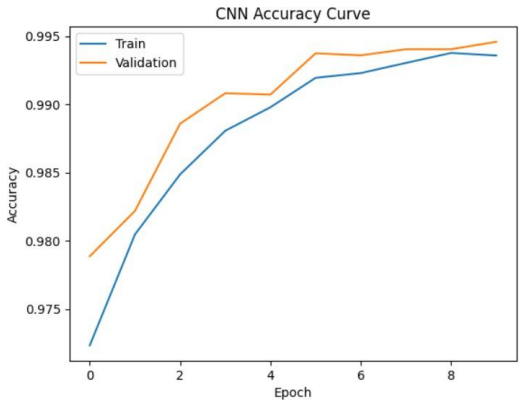


Fig. 5. Training and Validation Accuracy Curve of CNN Model

The learning curve shown in Fig. 5 illustrates illustrates the training and validation accuracy across epochs, demonstrating that the model converges effectively without significant over-fitting. The results show that the CNN model performs well for both classes with balanced performance across precision and recall values.

TABLE IV: CONFUSION MATRIX OF CNN MODEL

| Actual / Predicted | Normal | Attack |
|---|---|---|
| Normal | 13399 | 70 |
| Attack | 92 | 11634 |

From the confusion matrix, it can be observed that 13,399 normal traffic instances and 11,634 attack

instances were correctly classified. A small number of samples were misclas-sified, where 70 normal instances were incorrectly predicted as attacks, and 92 attack instances were incorrectly classified as normal traffic. Although the CNN model achieved slightly lower accuracy compared to the Random Forest model, it demonstrates strong capability in detecting complex patterns in network traffic data. The results indicate that CNN can effectively learn feature representations and identify intrusion activities with high accuracy.

*E. LSTM Model Results*

The Long Short Term Memory (LSTM) model was imple-mented to capture temporal dependencies and sequential pat-terns in network traffic data. LSTM networks are particularly suitable for intrusion detection because many cyber attacks occur in sequential stages and exhibit time dependent behavior.

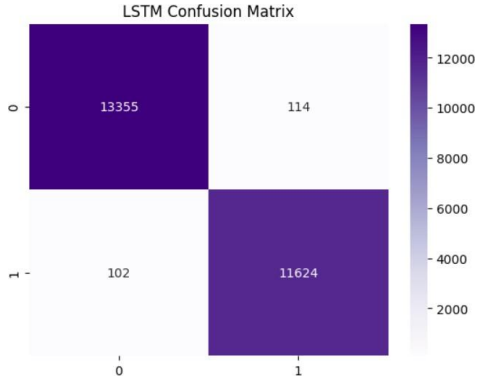


Fig. 6. Confusion Matrix of LSTM Model

During training, the LSTM model was trained for 10 epochs using the preprocessed NSL KDD dataset. The training accu-racy improved progressively from 96.51% in the first epoch to approximately 98.95% in the final epoch. Similarly, the vali-dation accuracy increased steadily and reached approximately 99.09%, indicating that the model generalized well to unseen data.

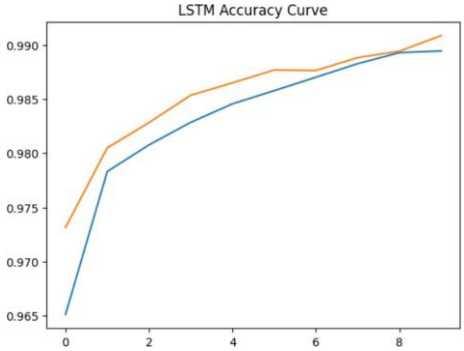


Fig. 7. Training and Validation Accuracy Curve of LSTM Model

The final evaluation on the testing dataset produced an overall accuracy of approximately 99.14%, demonstrating that the LSTM model effectively learns temporal patterns in network traffic and accurately distinguishes between normal and malicious activities. The training and validation accuracy curves shown in Fig. 7 illustrate the learning behavior of the model across training epochs. The curves indicate stable convergence with no significant signs of overfitting.

TABLE V: CONFUSION MATRIX OF LSTM MODEL

| Actual / Predicted | Normal | Attack |
|---|---|---|
| Normal | 13355 | 114 |
| Attack | 102 | 11624 |

*F. Hybrid CNN-LSTM Model Results*

The Hybrid CNN–LSTM model was implemented to com-bine the advantages of both Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) networks. In this architecture, CNN layers are responsible for extracting spatial features from network traffic data, while the LSTM layer captures temporal dependencies and sequential patterns. This hybrid approach enables the model to effectively detect complex and multi stage cyber attacks.

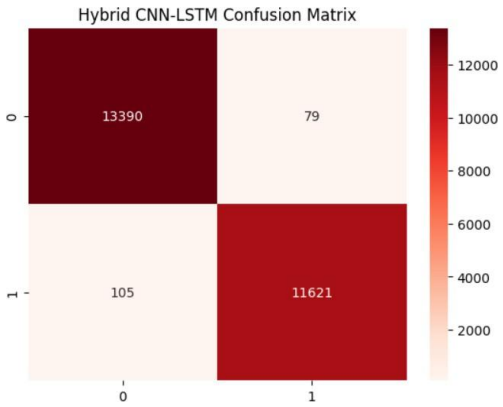


Fig. 8. Training and Validation Accuracy of Hybrid CNN-LSTM Model

During training, the Hybrid CNN–LSTM model was trained for 10 epochs using the preprocessed NSL KDD dataset. The training accuracy increased steadily from 94.80% in the first epoch to approximately 99.10% in the final epoch, while the validation accuracy reached approximately 99.25%, indicating stable learning and good generalization performance. The final evaluation on the testing

dataset achieved an overall accuracy of approximately 99.27%, which is the highest among all the evaluated models. This demonstrates the effectiveness of the hybrid architecture in detecting intrusion patterns within network traffic.

TABLE VI: CONFUSION MATRIX OF HYBRID CNN-LSTM MODEL

| Actual / Predicted | Normal | Attack |
|---|---|---|
| Normal | 13390 | 79 |
| Attack | 105 | 11621 |

From the confusion matrix, it can be observed that 13,390 normal traffic instances and 11,621 attack instances were correctly classified by the model. A small number of samples were misclassified, where 79 normal instances were incorrectly predicted as attacks, and 105 attack instances were incorrectly classified as normal traffic. These results demonstrate that the Hybrid CNN–LSTM model provides high intrusion detection accuracy with minimal misclassification, outperforming the individual CNN and LSTM models. By combining spatial and temporal feature learning, the hybrid model is capable of identifying complex intrusion patterns and dynamic attack behaviors, making it highly suitable for adaptive intrusion detection systems.

### *G. Model Comparison*

To evaluate the effectiveness of different intrusion detection models, a comparative analysis was conducted using the De-cision Tree, Random Forest, CNN, LSTM, and Hybrid CNN LSTM models. The performance of each model was measured using classification accuracy on the NSL KDD testing dataset.

TABLE VII: ACCURACY COMPARISON OF DIFFERENT IDS MODELS

| Model | Accuracy |
|---|---|
| Decision Tree | 0.997023 |
| Random Forest | 0.998015 |
| CNN | 0.993570 |
| LSTM | 0.991427 |
| Hybrid CNN-LSTM | 0.992697 |

The performance of different intrusion detection models was evaluated using the NSL-KDD testing dataset. The models considered in this study include Decision Tree, Random For-est, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Hybrid CNN-LSTM. The accuracy values obtained for each model are presented in Fig. X.

The comparison results indicate that all models achieved high classification accuracy, with performance values above 99%. Among the evaluated models, the Random Forest clas-sifier achieved the highest accuracy of 0.998015 (99.80%), followed by the Decision Tree model with an accuracy of 0.997023 (99.70%). The deep learning models also demon-strated strong performance, where the CNN model achieved an accuracy of 0.993570 (99.36%), the Hybrid CNN-LSTM model achieved 0.992697 (99.27%), and the LSTM model achieved 0.991427 (99.14%).

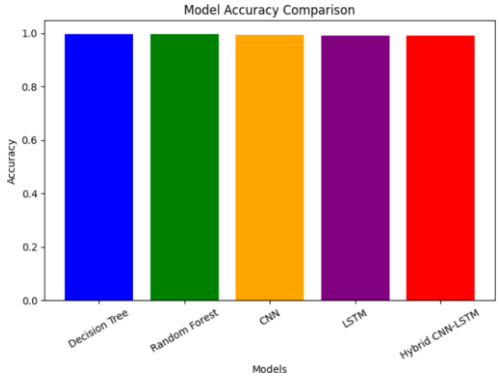


Fig. 9. Accuracy Comparison of Different Intrusion Detection Models

Although traditional machine learning models such as De-cision Tree and Random Forest achieved slightly higher ac-curacy values, deep learning models are capable of learning complex spatial and temporal relationships present in network traffic data. These models are particularly useful for detecting multi-stage and sophisticated cyber attacks.

Overall, the comparison results demonstrate that both ma-chine learning and deep learning models provide strong in-trusion detection capability. Integrating these models within an adaptive intrusion detection framework allows the system to utilize the strengths of each model under different network conditions.

### *H. Adaptive IDS Model Selection under Dynamic Traffic Con-ditions*

To further evaluate the effectiveness of the proposed system, the Adaptive Intrusion Detection System (Adaptive IDS) was tested under simulated dynamic network traffic conditions. The system dynamically selects the most suitable model based on the

characteristics of incoming network traffic.

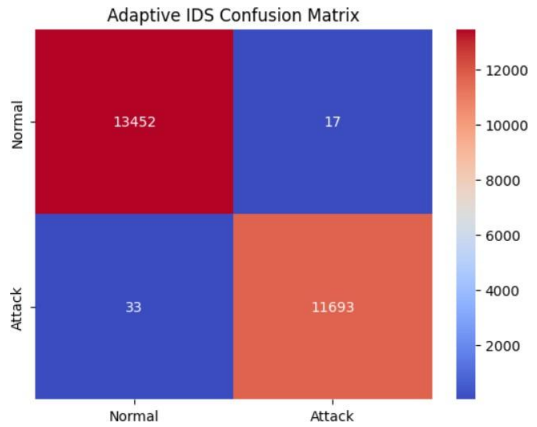


Fig. 10. Model Selection Frequency of the Adaptive IDS under Dynamic Network Traffic Conditions

The above figure shows the frequency of model selection by the adaptive intrusion detection system during dynamic traffic simulations. The Hybrid CNN-LSTM model was selected most frequently, particularly under attack-dominant traffic conditions where complex intrusion patterns occur. The Random Forest model was selected during mixed traffic scenarios, while the Decision Tree model was selected during normal traffic conditions due to its fast classification capability.

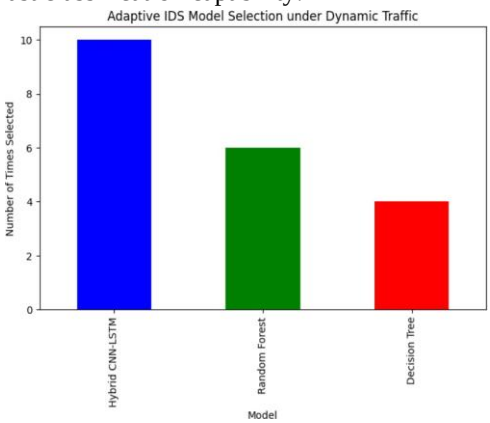


Fig. 11. Adaptive Intrusion Detection System Result

The confusion matrix shown in Fig. 11 demonstrates the classification performance of the adaptive IDS. The system correctly classified 13,452 normal traffic instances and 11,693 attack instances. Only 17 normal samples were incorrectly pre-dicted as attacks, while 33 attack samples were misclassified as normal traffic. These results indicate that the proposed adaptive intrusion detection system achieves high detection accuracy with very low false positive and false negative rates.

TABLE VIII: CONFUSION MATRIX OF ADAPTIVE IDS

| Actual / Predicted | Normal | Attack |
|---|---|---|
| Normal | 13452 | 17 |
| Attack | 33 | 11693 |

From the confusion matrix, it can be observed that 13,452 normal traffic instances and 11,693 attack instances were correctly classified. Only 17 normal samples were incorrectly predicted as attacks, and 33 attack samples were misclassified as normal traffic. These results indicate that the proposed Adaptive IDS achieves high detection accuracy while maintaining very low false positive and false negative rates. The results demonstrate that the adaptive model selection strategy effectively improves intrusion detection performance by dynamically utilizing the most suitable model for different network traffic conditions. This capability enables the system to maintain robust detection performance in dynamic network environments, making it suitable for real world intrusion detection applications.

### *I. Discussion*

The experimental findings indicate that the proposed Traffic-Aware Adaptive Intrusion Detection System delivers high detection accuracy across both machine learning and deep learning models. Conventional machine learning techniques such as Decision Tree and Random Forest achieve strong classification performance along with faster training time. Deep learning approaches, including CNN, LSTM, and the Hybrid CNN-LSTM model, are effective in capturing complex spatial and temporal relationships within network traffic data. In addition, the adaptive model selection mechanism enhances the overall robustness of the system by dynamically choosing the most appropriate model based on varying network traffic conditions.

## V. CONCLUSION

In this work, an Adaptive Intrusion Detection System (Adaptive IDS) was developed to improve network security under dynamic traffic conditions. The proposed framework integrates both machine learning and deep learning techniques, including Decision Tree, Random Forest, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and a Hybrid CNN-LSTM model. These models were trained and tested using the

NSL-KDD dataset, which is widely recognized in intrusion detection research.The experimental results show that all the implemented models achieved high accuracy in classifying both normal and malicious network traffic. Among them, the Decision Tree and Random Forest models provided the highest accuracy, while the deep learn-ing models demonstrated strong ability in capturing complex patterns within network data. The Hybrid CNN-LSTM model, in particular, effectively combines spatial and temporal feature learning, making it suitable for detecting advanced intrusion patterns.To further improve performance, an adaptive model selection mechanism was introduced. The proposed system dynamically selects the most suitable model based on current traffic conditions. Specifically, the Decision Tree model is applied in normal traffic scenarios, Random Forest is used for mixed traffic conditions, and the Hybrid CNN-LSTM model is selected during attack-dominant situations. This adaptive ap-proach helps maintain consistent detection performance across varying network environments.Overall, the results confirm that the proposed Adaptive IDS achieves high detection accuracy, reduced false alarm rates, and improved robustness. Therefore, it can be considered an effective solution for modern intrusion detection systems and can contribute to strengthening cyber-security in real-world network environments.

## VI. FUTURE WORK

Although the proposed Adaptive Intrusion Detection System (Adaptive IDS) demonstrates high accuracy and strong performance in detecting network intrusions, several improvements can be explored in future research. First, the proposed system was evaluated using the NSL KDD dataset, which is a widely used benchmark dataset for intrusion detection research. Future work can focus on evaluating the system using more recent and realistic network traffic datasets, such as CICIDS2017 and UNSW NB15, to further analyze its performance in real world network environments. Second, the current adaptive IDS selects detection models based on pre-defined traffic conditions. In future research, more advanced intelligent model selection techniques, such as reinforcement learning or meta learning, can be implemented to automatically determine the most suitable model based on real time network traffic characteristics. Third, the proposed framework can be extended by incorporating additional advanced deep learning architectures, such as Transformer based models or Graph Neural Networks (GNNs), which may further improve the detection of sophisticated cyber attacks and complex network behaviors. Finally, the system can be deployed in a real time network monitoring environment to evaluate its performance in practical cybersecurity applications. Integration with software defined networking(SDN) or cloud based security platforms could further enhance the scalability, adaptability, and real time detection capabilities of the intrusion detection system.

## REFERENCES


[1] W. Liu, “Dynamic network intrusion detection model based on trans-former and adversarial autoencoder,” *International Journal of Intelligent Networks*, 2025, doi: 10.1016/j.ijin.2025.11.002.

[2] W. Villegas-Ch, J. Govea, R. Gutierrez, A. Maldonado Navarro, and A. Mera-Navarrete, “Effectiveness of an adaptive deep learning-based intrusion detection system,” *IEEE Access*, vol. 12, pp. 184010–184023, 2024, doi: 10.1109/ACCESS.2024.3512363.

[3] U. Shahid, M. Z. Hussain, M. Z. Hasan, A. Haider, J. Ali, and J. Altaf, “Hybrid intrusion detection system for RPL IoT networks using machine learning and deep learning,” *IEEE Access*, vol. 12, pp. 113099–113112, 2024, doi: 10.1109/ACCESS.2024.3442529.

[4] J. Du, K. Yang, Y. Hu, and L. Jiang, “NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning,” *IEEE Access*, 2023, doi: 10.1109/ACCESS.2023.3254915.

[5] H. C. Altunay and Z. Albayrak, “A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks,” *Engineering Science and Technology, an International Journal*, vol. 38, Art. no. 101322, 2023, doi: 10.1016/j.jestch.2022.101322.

[6] I. A. Abdulmajeed and I. M. Husien, “MLIDS22 – IDS design by applying hybrid CNN-LSTM model on mixed-datasets,” *Informatica*, vol. 46, no. 8, pp. 121–134, 2022, doi: 10.31449/inf.v46i8.4348.

[7] Y. Wang, “Deep learning–based network

intrusion detection systems," in *Proc. 2nd Int. Conf. Machine Learning and Automation*, 2024, doi: 10.54254/2755-2721/109/2024.18104.

[8] P. Sinha, D. Sahu, S. Prakash, T. Yang, R. S. Rathore, and V. K. Pandey, "A high performance hybrid LSTM-CNN secure architecture for IoT environments using deep learning," *Scientific Reports*, vol. 15, Art. no. 9684, 2025, doi: 10.1038/s41598-025-94500-5.

[9] S. Chatterjee, S. Chaudhary, and A. K. Cherukuri, "Intrusion detection system using deep learning for network security," School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, India, 2024.

[10] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[11] J. Du, K. Yang, Y. Hu, and L. Jiang, "A CNN-LSTM based in-trusion detection model using KDD Cup99, NSL-KDD and UNSW-NB15 datasets," *IEEE Access*, vol. 11, 2023, doi: 10.1109/AC-CESS.2023.3254915.

[12] O. Belarbi, T. Spyridopoulos, E. Anthi, I. Mavromatis, P. Carnelli, and A. Khan, "Federated deep learning for intrusion detection in IoT networks," *arXiv preprint*, arXiv:2306.02715, 2023.

[13] M. Cantone, C. Marrocco, and A. Bria, "Cross-dataset evaluation of machine learning models for intrusion detection using CIC and LycoS datasets," *IEEE Access*, vol. 11, 2023, doi: 10.1109/AC-CESS.2024.3472907.

[14] N. Xing, S. Zhao, Y. Wang, K. Ning, and X. Liu, "A dynamic intrusion detection system capable of detecting unknown attacks," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 7, pp. 391–400, 2023.

[15] J. Jabez, A. Alkhayyat, G. Vengatesan, and R. Vasanthan, "Adaptive intrusion detection mechanisms for enhancing security in cloud-hosted big data systems," *Frontiers in Health Informatics*, vol. 13, no. 3, pp. 5313–5327, 2024.

[16] A. A. Aliyu, J. Liu, and E. Gilliard, "A decentralized and self-adaptive intrusion detection approach using continuous learning and blockchain technology," *Journal of Data Science and Intelligent Systems*, Oct. 2024, doi: 10.47852/bonviewJDSIS42023803.

[17] H. G. N. and A. H. S., "An enhanced network intrusion detection system using ADASYN and hybrid residual block techniques," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 25082–25087, 2025.

[18] M. Cantone, C. Marrocco, and A. Bria, "Machine learning in network intrusion detection: A cross-dataset generalization study," *IEEE Access*, vol. 11, 2024, doi: 10.1109/ACCESS.2024.3472907.

[19] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi-class SVM," *Journal of King Saud University – Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017.

[20] A. K. Cherukuri, S. T. Ikram, G. Li, and X. Liu, "Artificial intelligence-based approaches for anomaly detection," in *Encrypted Network Traffic Analysis*. Cham, Switzerland: Springer, 2024, pp. 73–99.

[21] V. D. Gowda, D. Palanikkumar, S. Dekka, K. D. V. Prasad, and S. Singh, "Future trends in V2X communication and interoperability," in *Digital Convergence in Intelligent Mobility Systems*. John Wiley & Sons, Inc., Aug. 2025, pp. 217–239.

[22] A. Kumar, B. Ashreetha, A. M. Reddy, S. Dekka, D. Gowda, and K. D. V. Prasad, "Maximizing energy efficiency in wireless sensor networks for IoT with advanced techniques and solutions," in *Proc. 5th International Conference on Pervasive Computing and Social Networking (ICPCSN)*. IEEE, May 2025, pp. 448–454.

[23] S. Dekka, B. Sambana, K. N. Raju, D. M. Sai, M. Pallavi, and K. K. Reddy, "Performance evaluation of QoS in MAODV routing protocol in MANETS," in *Advances in Machine Learning and Big Data Analytics I (ICMLBDA 2023)*, vol. 441. Springer Nature, Feb. 2025, pp. 133–150.

[24] S. Dekka, K. N. Raju, D. M. Sai, M. Pallavi, and B. Sambana, "Minimize the energy consumption to increase the network lifetime for green IoT environment," in *Advances in*

*Machine Learning and Big Data Analytics I (ICMLBDA 2023)*, vol. 441. Springer Nature, Feb. 2025, pp. 151–159.

[25] V. D. Gowda, M. S. P. A. Mary, A. Sharma, and S. Dekka, "Deep learning approaches for real-time data analytics in IoT sensor networks," in *Universal Threats in Expert Applications and Solutions*, vol. 2. Springer Nature, Nov. 2024, pp. 239–248.