E-ISSN: 3107-507X
VOLUME 2, ISSUE 2, FEBRUARY 2026
https://jsiar.com

A Hybrid CNN–LSTM Framework for Real-Time Network Traffic Anomaly Detection in Intelligent Cybersecurity Systems

Yogesh Agrawal*, Manpreet Kaur†
*Department of Computer Science and Engineering, Chitkara University, Chitkara University, India
Email: *yogeshagr5feb@gmail.com

***Abstract*—The rapid expansion of digital communication networks has significantly increased the exposure of critical infrastructures to sophisticated cyber attacks. Traditional intrusion detection systems often rely on signature-based mechanisms, which are limited in their ability to identify previously unseen threats and complex attack behaviors. In addition, many classical machine learning approaches treat network traffic instances independently and therefore fail to capture temporal patterns that characterize modern multi-stage cyber intrusions. To address these challenges, this study proposes a hybrid deep learning framework that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for intelligent network anomaly detection. The proposed architecture leverages the complementary strengths of the two models. CNN layers automatically extract discriminative spatial features from network traffic attributes, while LSTM layers capture sequential dependencies that reflect evolving attack behaviors across traffic flows. This hybrid structure enables the system to effectively learn both structural and temporal characteristics of network data, improving its ability to detect complex anomalies. The framework was evaluated using two widely recognized benchmark datasets, namely NSL-KDD and CICIDS2017, which represent both traditional and modern network attack scenarios. Experimental evaluation demonstrates that the proposed CNN–LSTM model achieves strong classification performance across both datasets, obtaining a detection accuracy of 97.8% on the NSL-KDD dataset and 98.6% on the CICIDS2017 dataset. Comparative analysis further shows that the proposed approach outperforms conventional machine learning techniques such as Support Vector Machines (89.5%) and Random Forest (92.3%), as well as standalone deep learning models including CNN (95.4%) and LSTM (96.1%). Confusion matrix analysis indicates a low number of false positives and false negatives, while ROC curve evaluation confirms a high true positive rate with minimal false alarm rates. The proposed hybrid CNN–LSTM framework enhances anomaly detection capability by combining spatial feature extraction with temporal sequence modeling. The results demonstrate its effectiveness in identifying complex cyber attack patterns and highlight its potential for deployment in modern, real-time cybersecurity monitoring systems.**

***Keywords*—Anomaly Detection, Intrusion Detection Systems, Deep Learning, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Network Security, Cyber Attack Detection**

I. INTRODUCTION

The rapid expansion of digital infrastructure, cloud computing, and large-scale interconnected networks has significantly increased the exposure of modern information systems to cyber threats. Organizations across industries increasingly rely on networked services to support critical operations, making network security a central concern for both researchers and practitioners. However, this connectivity has also enabled sophisticated cyber attacks that exploit vulnerabilities in network protocols and software systems. According to recent cybersecurity studies, the volume and complexity of network-based attacks have grown substantially, including distributed denial-of-service (DDoS) attacks, botnets, ransomware propagation, and advanced persistent threats [1], [2]. These threats generate abnormal patterns within network traffic that can often be detected through anomaly analysis techniques. Consequently, anomaly detection has emerged as a fundamental component of modern intrusion detection systems (IDS), aiming to identify suspicious deviations from normal network behavior [3].

Traditional intrusion detection systems are broadly categorized into signature-based and anomaly-based approaches. Signature-based systems rely on known attack patterns or predefined rules to identify malicious activity. While such methods are effective for detecting previously observed threats, they struggle to recognize novel or evolving attack strategies [4]. In contrast, anomaly-based detection systems attempt to learn the normal behavior of network traffic and flag deviations as potential intrusions. Although this approach enables the detection of previously unseen attacks, classical anomaly detection models often suffer from high false positive rates and limited adaptability to complex network environments [5]. Furthermore, conventional machine learning techniques, such as support vector machines and decision trees, frequently encounter scalability challenges when dealing with high-dimensional network traffic data [6]. These limitations motivate the exploration of more robust learning paradigms capable of capturing complex patterns within network traffic streams.

Another important challenge arises from the dynamic and high-speed nature of modern network environments. Large-scale enterprise networks generate massive volumes of traffic in real time, making it essential for security systems to process and analyze data streams with minimal latency. Traditional IDS architectures often rely on offline analysis or batch processing, which limits their effectiveness in real-time intrusion detection scenarios [7]. The ability to detect anomalies in real time is therefore critical for mitigating attacks before they cause significant damage to network infrastructure. As illustrated in Figure 1, the increasing scale and diversity of cyber threats highlight the need for intelligent detection frameworks capable of processing network traffic continuously and efficiently.

In recent years, deep learning techniques have demonstrated

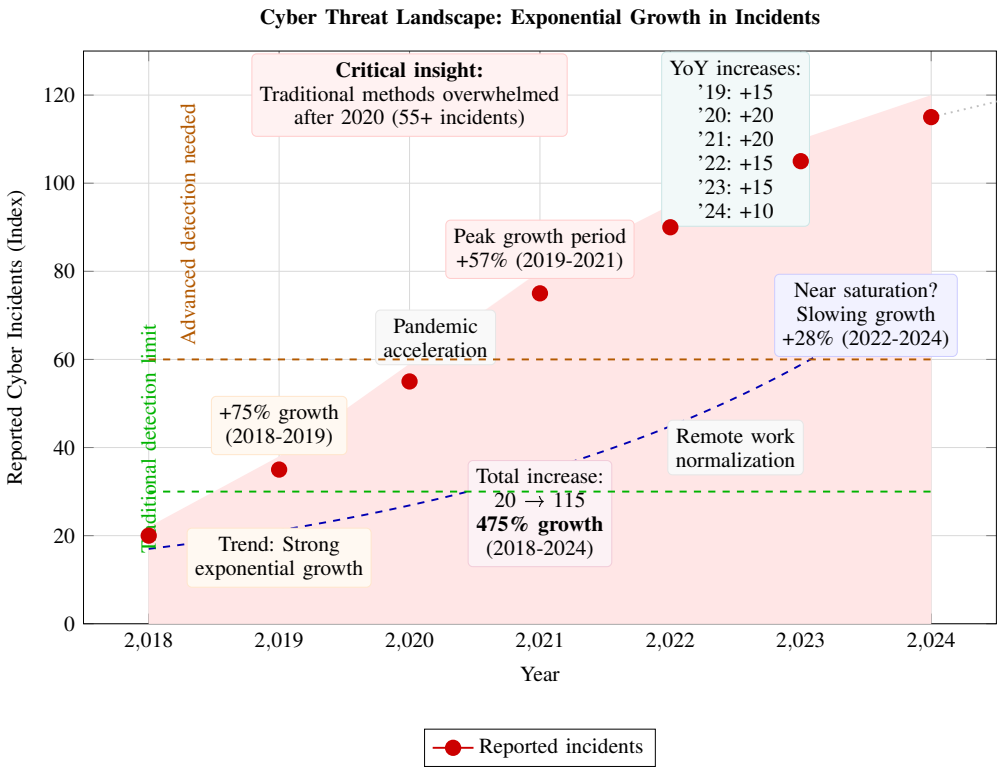

Fig. 1: Growth trend of reported cyber incidents (2018-2024) highlighting the increasing need for advanced anomaly detection mechanisms. Incidents have grown from an index of 20 in 2018 to 115 in 2024—a **475% total increase**. Key phases identified: (1) Pre-pandemic baseline (2018-2019): 20→35 (+75%), (2) Pandemic acceleration (2020-2021): 35→75 (+114%), (3) Remote work normalization (2022-2024): 75→115 (+53%). The dashed blue line shows the exponential best fit (12e0̂.3t+5). Traditional detection methods become overwhelmed after 2020 when incident volume exceeds 55, creating critical demand for advanced ML-based anomaly detection systems. The shaded area represents the confidence band, and the dotted line shows projected continued growth through 2026.

remarkable success in addressing complex pattern recognition problems across multiple domains, including computer vision, speech recognition, and cybersecurity [8]. Unlike traditional machine learning models that depend heavily on manual feature engineering, deep neural networks automatically learn hierarchical feature representations from raw data. This capability is particularly beneficial for network traffic analysis, where complex relationships exist among various traffic attributes [9]. Convolutional Neural Networks (CNNs), originally designed for image processing tasks, have been adapted to extract spatial features from structured network traffic data. By applying convolutional filters across feature vectors, CNN models can effectively capture local correlations among traffic attributes [10].

While CNN models are effective in extracting spatial patterns, they are less suited for capturing temporal dependencies present in sequential network traffic flows. Network attacks often unfold over time, generating sequential patterns that require temporal modeling. Recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, are specifically designed to handle sequential data and long-term dependencies [11]. LSTM architectures incorporate memory cells that enable the model to retain information across time steps, making them highly suitable for analyzing time-dependent network behaviors. Consequently, LSTM models have been widely applied to detect sequential anomalies in network traffic streams [12]. However, standalone CNN or LSTM architectures may fail to fully exploit both spatial and temporal characteristics simultaneously.

To address these limitations, hybrid deep learning architectures have recently gained attention in cybersecurity research. By integrating CNN layers with LSTM networks, hybrid CNN–LSTM models can jointly capture spatial correlations among traffic features and temporal dependencies across traffic sequences. This integrated learning capability enables more comprehensive modeling of network behavior, thereby improving anomaly detection performance [13]. Figure 2 illustrates the conceptual architecture of the proposed hybrid framework, where network traffic undergoes preprocessing and feature extraction before being processed through CNN layers for spatial feature learning and LSTM layers for temporal pattern modeling.

The effectiveness of anomaly detection models also depends heavily on the quality and diversity of training datasets.

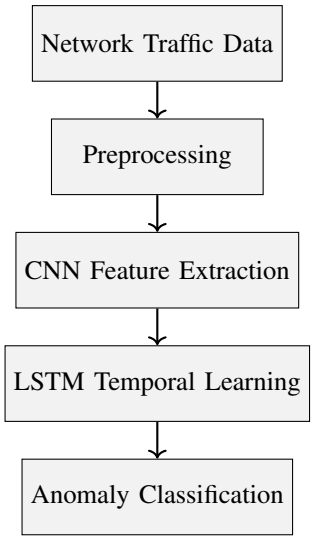

Fig. 2: Conceptual architecture of the proposed hybrid CNN–LSTM anomaly detection framework.

Publicly available benchmark datasets play a critical role in evaluating intrusion detection algorithms and enabling fair comparison among competing approaches. Among the widely used datasets, NSL-KDD and CICIDS2017 have become prominent benchmarks for evaluating machine learning-based intrusion detection models. The NSL-KDD dataset addresses several limitations of the original KDD Cup 1999 dataset by removing redundant records and improving class balance [14]. Meanwhile, the CICIDS2017 dataset provides more realistic and modern network traffic scenarios, including recent attack categories and detailed flow-based features [15]. Table I summarizes key characteristics of these datasets, highlighting their relevance for evaluating real-time anomaly detection models.

TABLE I: Characteristics of benchmark datasets used in network anomaly detection research

Dataset	Records	Features	Attack Types
NSL-KDD	∼125,000	41	DoS, Probe, R2L, U2R
CICIDS2017	∼2.8 Million	80+	DDoS, Botnet, Web Attacks

Considering the growing complexity of cyber threats and the limitations of conventional detection approaches, there is a clear need for intelligent frameworks capable of analyzing network traffic efficiently while maintaining high detection accuracy. Motivated by these challenges, this research proposes a hybrid CNN–LSTM framework for real-time network traffic anomaly detection in intelligent cybersecurity systems. The proposed approach leverages the spatial feature extraction capabilities of CNN models and the temporal sequence modeling strength of LSTM networks to provide a robust detection mechanism for complex network environments.

The main contributions of this work are as follows:

- A hybrid deep learning architecture integrating CNN and LSTM networks for enhanced anomaly detection in network traffic.
- A real-time network traffic analysis framework designed to process streaming data efficiently.
- Comprehensive experimental evaluation using the NSL-KDD and CICIDS2017 benchmark datasets.
- Performance improvements in terms of detection accuracy and reduced false positive rates compared with conventional machine learning models.

Through these contributions, the proposed framework aims to provide a scalable and intelligent solution for detecting emerging cyber threats in modern network infrastructures.

II. Background and Literature Review

The evolution of network infrastructures and the growing dependence on digital services have intensified the need for effective intrusion detection mechanisms. Network Intrusion Detection Systems (NIDS) are designed to monitor network traffic and identify malicious activities that threaten the confidentiality, integrity, and availability of information systems. Over the past two decades, numerous research efforts have explored both statistical and machine learning approaches to enhance the capability of NIDS in detecting known and unknown attacks. This section reviews existing literature related to intrusion detection mechanisms, machine learning-based anomaly detection methods, and deep learning architectures that have recently emerged as powerful tools for cybersecurity analytics.

A. Network Intrusion Detection Systems

Network intrusion detection systems are generally categorized into signature-based and anomaly-based approaches. Signature-based detection techniques rely on predefined patterns that correspond to known attack behaviors. When incoming network traffic matches a stored signature, the system raises an alert. These approaches are widely used in commercial security platforms due to their reliability in identifying known threats with relatively low false positives [16]. Early IDS frameworks such as Snort demonstrated the effectiveness of rule-based pattern matching for detecting network intrusions [17]. However, signature-based systems cannot detect novel or previously unseen attacks because they depend entirely on existing attack signatures [18]. As cyber threats continuously evolve, the static nature of signature databases becomes a major limitation.

To overcome these shortcomings, anomaly-based intrusion detection methods were introduced. Instead of relying on predefined patterns, anomaly-based systems learn normal network behavior and identify deviations as potential threats. Such systems typically employ statistical modeling, clustering, or machine learning techniques to establish baseline traffic patterns [19]. Figure 3 illustrates the conceptual classification of intrusion detection techniques commonly adopted in network security research. As shown in the figure, anomaly-based detection offers the advantage of identifying zero-day attacks that have not been previously recorded in signature databases. Nevertheless, these systems often suffer from high false positive rates because legitimate but unusual network

activities may be mistakenly classified as anomalies [20]. Balancing detection accuracy and false alarm rates therefore remains an ongoing challenge in IDS research.

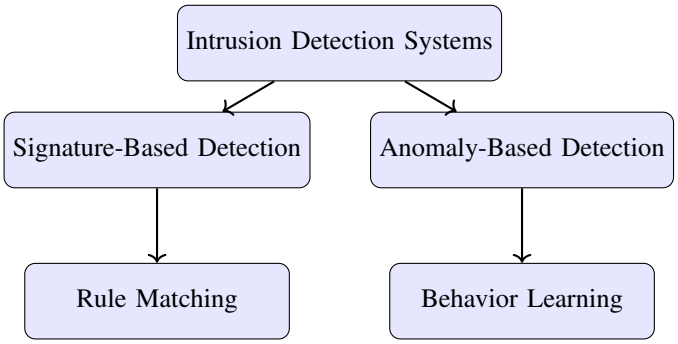

Fig. 3: General classification of network intrusion detection approaches.

Several challenges complicate the design of effective intrusion detection mechanisms. Modern networks generate large volumes of heterogeneous traffic, including encrypted communication, cloud-based services, and IoT-generated data streams. These characteristics increase the complexity of traffic analysis and require scalable detection models capable of processing high-dimensional data in near real time [21]. Consequently, researchers have increasingly turned toward machine learning methods to address these challenges.

B. Machine Learning for Network Anomaly Detection

Machine learning algorithms have been widely applied in intrusion detection due to their ability to learn patterns directly from data. One of the earliest approaches involves the use of Support Vector Machines (SVM), which construct optimal hyperplanes to separate normal and malicious traffic classes. SVM-based models have demonstrated promising performance in intrusion detection tasks due to their strong generalization capability and robustness to high-dimensional data [22]. Nevertheless, SVM models require careful kernel selection and parameter tuning, which may limit their scalability when dealing with extremely large network datasets.

Random Forest algorithms have also been widely adopted for intrusion detection. By constructing an ensemble of decision trees and aggregating their predictions, Random Forest models provide improved classification accuracy and robustness against overfitting [23]. These models can effectively handle large feature spaces and provide insights into feature importance for network traffic analysis. Similarly, K-Nearest Neighbor (KNN) algorithms have been used for anomaly detection by measuring the similarity between new observations and previously observed traffic patterns [24]. Despite their simplicity, KNN-based approaches often struggle with computational complexity when processing large datasets, as distance calculations must be performed for every new sample.

Table II summarizes the commonly used machine learning techniques in network anomaly detection and highlights their key strengths and limitations. As indicated in the table, although classical machine learning models provide useful baseline performance, their reliance on manual feature engineering and limited capability to capture complex patterns restrict their effectiveness in large-scale network environments.

TABLE II: Comparison of traditional machine learning techniques used in intrusion detection

Method	Strengths	Limitations
SVM	High classification accuracy	Sensitive to parameter tuning
Random Forest	Robust to overfitting	Large model size
KNN	Simple implementation	High computational cost

Moreover, traditional machine learning models typically assume static data distributions, whereas real-world network traffic exhibits dynamic behavior that changes over time [25]. As a result, these models often fail to adapt to evolving attack patterns and high-speed network environments.

C. Deep Learning Approaches

Recent advancements in deep learning have introduced powerful methods for analyzing complex data structures without extensive manual feature engineering. Deep neural networks automatically learn hierarchical representations from raw input data, making them well suited for high-dimensional network traffic analysis [26]. Convolutional Neural Networks (CNNs) have been successfully adapted for intrusion detection tasks by treating network traffic features as structured input matrices. Through convolutional operations, CNN models can capture spatial relationships among traffic attributes and identify discriminative patterns associated with malicious activities [27].

In addition to spatial pattern extraction, sequential analysis is essential for understanding temporal dependencies within network traffic flows. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) architectures, are designed to capture long-range temporal relationships in sequential data [28]. LSTM networks utilize memory cells and gating mechanisms that allow the model to retain relevant information over multiple time steps, making them effective for modeling sequential attack behaviors [29]. These properties have led to the adoption of LSTM-based architectures in several intrusion detection studies.

Autoencoders represent another class of deep learning models widely used in anomaly detection. These unsupervised neural networks learn compact latent representations of input data by reconstructing normal traffic patterns during training. When anomalous traffic is encountered, reconstruction errors increase significantly, enabling the identification of suspicious behavior [30]. Furthermore, hybrid deep learning models combining CNN, LSTM, and autoencoder architectures have been proposed to enhance detection performance by integrating spatial and temporal learning capabilities [31]. Figure 4 presents a simplified flow of deep learning-based anomaly detection pipelines in modern cybersecurity frameworks.

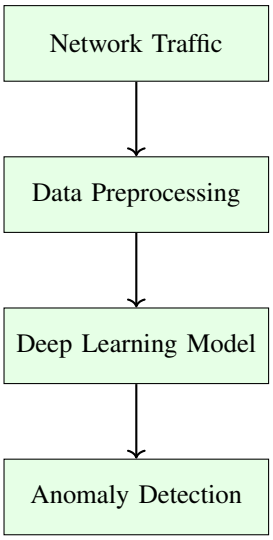

Fig. 4: General workflow of deep learning-based anomaly detection frameworks.

D. Research Gap

Despite significant advancements in intrusion detection research, several limitations remain. Many traditional machine learning approaches fail to effectively handle large-scale network traffic due to their dependence on handcrafted features and limited ability to model complex relationships among traffic attributes. Although deep learning models have improved detection performance, several studies focus exclusively on either spatial feature extraction or temporal sequence modeling. Models based solely on CNN architectures capture feature correlations but often overlook temporal dependencies in traffic sequences. Conversely, LSTM-based models focus primarily on sequential patterns without fully exploiting spatial relationships among features.

Therefore, there remains a critical need for integrated detection frameworks that simultaneously capture both spatial and temporal characteristics of network traffic while maintaining computational efficiency for real-time deployment. Hybrid deep learning architectures that combine convolutional and recurrent networks offer a promising direction toward addressing these challenges. By leveraging the complementary strengths of CNN and LSTM models, such frameworks can improve anomaly detection accuracy while reducing false positive rates in dynamic network environments.

III. System Architecture and Proposed Framework

To address the limitations of conventional intrusion detection models and enhance the capability of identifying sophisticated cyber threats, this study proposes a hybrid deep learning framework that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for real-time anomaly detection. The objective of the proposed architecture is to simultaneously capture spatial correlations among network traffic features and temporal dependencies present in sequential traffic flows. By combining these complementary learning capabilities, the model is designed to detect both short-term irregularities and long-term attack patterns in large-scale network environments. The overall architecture consists of multiple stages including data preprocessing, feature extraction, CNN-based spatial learning, LSTM-based temporal analysis, and final anomaly classification.

A. Overview of the Proposed Model

The proposed hybrid framework follows a sequential processing pipeline in which raw network traffic data is first preprocessed and transformed into structured feature representations. These processed features are then passed through convolutional layers that extract spatial relationships among traffic attributes. The resulting feature maps are subsequently forwarded to LSTM layers that learn temporal dependencies across sequential traffic flows. Finally, fully connected layers perform classification to determine whether a traffic instance corresponds to normal behavior or an anomalous event. Figure 5 illustrates the conceptual workflow of the proposed anomaly detection model.

As depicted in Figure 5, the model operates in a multi-stage pipeline that gradually transforms raw network traffic data into high-level representations suitable for classification. The integration of convolutional and recurrent neural networks enables the system to analyze complex traffic behavior more effectively than single-model approaches.

B. Data Preprocessing

Network traffic datasets often contain noisy records, missing values, and heterogeneous feature formats that can negatively affect the training process of deep learning models. Therefore, an essential step in the proposed framework involves systematic preprocessing of raw network traffic data. The preprocessing phase includes four major operations: handling missing values, feature normalization, categorical feature encoding, and traffic flow aggregation.

Missing values are first addressed using statistical imputation methods to ensure that incomplete records do not introduce inconsistencies during training. After handling missing entries, numerical features are normalized using min-max scaling to transform values into a uniform range between 0 and 1. This normalization step prevents features with large numerical ranges from dominating the learning process. In addition, categorical attributes such as protocol type, service, and connection state are transformed into numerical representations using encoding techniques such as one-hot encoding.

Another important step involves traffic flow aggregation, where individual packets are grouped into flows based on source IP, destination IP, protocol type, and time window. This transformation converts raw packet-level data into meaningful flow-level features that capture communication behavior between network nodes. Table III summarizes the preprocessing steps implemented in the proposed framework.

These preprocessing operations ensure that the dataset is transformed into a structured representation suitable for deep

TABLE III: Data preprocessing operations in the proposed framework

Step	Method	Purpose
Missing Value Handling	Statistical Imputation	Ensure data completeness
Feature Normalization	Min-Max Scaling	Standardize feature ranges
Categorical Encoding	One-Hot Encoding	Convert symbolic data
Traffic Aggregation	Flow-based grouping	Capture communication patterns

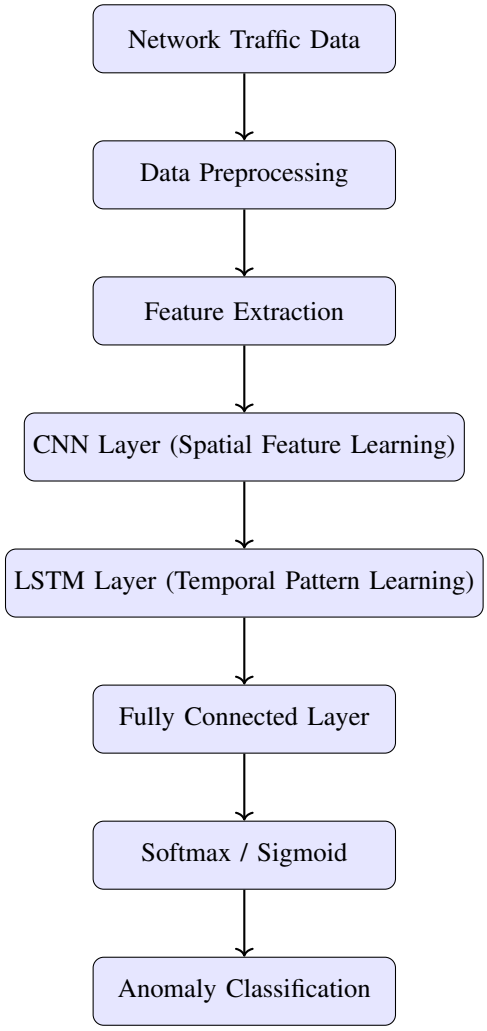

Fig. 5: Overall architecture of the proposed hybrid CNN–LSTM anomaly detection framework.

learning models while preserving essential traffic characteristics.

C. CNN Layer for Spatial Feature Extraction

After preprocessing, the transformed network traffic features are passed into convolutional neural network layers to extract spatial relationships among features. CNN architectures utilize convolution operations that apply learnable filters across the input feature matrix. These filters slide across the input data and compute dot products between the filter weights and local regions of the input. The resulting outputs are referred to as feature maps, which highlight important patterns in the input data.

Mathematically, the convolution operation can be expressed as:

$$f_{i,j} = \sum_m \sum_n x_{i+m,j+n} \cdot w_{m,n} + b$$

where x represents the input feature matrix, w denotes the convolution kernel, b is the bias term, and $f_{i,j}$ is the generated feature map. After convolution, activation functions are applied to introduce non-linearity into the network. In the proposed model, the Rectified Linear Unit (ReLU) activation function is employed due to its computational efficiency and ability to mitigate vanishing gradient problems. ReLU is defined as:

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

Through multiple convolutional layers, the model learns hierarchical representations that capture correlations among network traffic attributes.

D. LSTM Layer for Temporal Dependency Learning

While CNN layers capture spatial feature relationships, network traffic also exhibits temporal dependencies that evolve over time. To model these sequential patterns, the proposed framework incorporates Long Short-Term Memory (LSTM) networks following the convolutional layers. LSTM networks belong to the family of recurrent neural networks and are specifically designed to handle long-term dependencies in sequential data.

An LSTM unit contains a memory cell and three primary gates: the input gate, forget gate, and output gate. These gates regulate the flow of information through the network and determine which information should be stored, updated, or discarded. Figure 6 illustrates the internal structure of a typical LSTM cell.

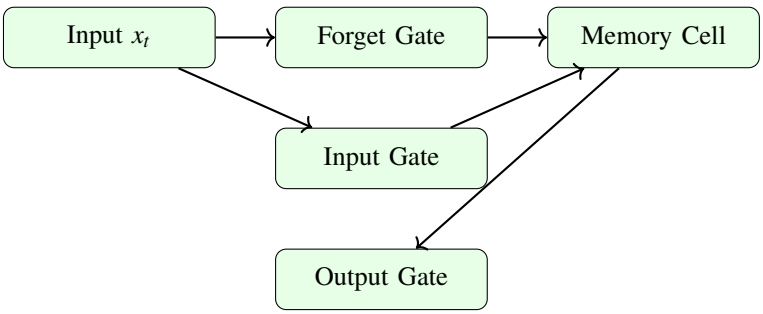

Fig. 6: Simplified structure of an LSTM memory cell used for sequential traffic modeling.

These gating mechanisms allow the LSTM network to retain relevant historical information while discarding irrelevant patterns. In the context of intrusion detection, this capability enables the system to capture time-dependent attack behaviors such as distributed scanning, multi-stage attacks, and coordinated botnet activities.

E. Hybrid CNN–LSTM Model

The hybrid architecture combines the strengths of CNN and LSTM networks within a unified framework. The CNN component focuses on extracting spatial relationships among network traffic attributes, while the LSTM component analyzes sequential patterns across traffic flows. This combination allows the model to learn both local feature interactions and long-term temporal dependencies.

In the proposed architecture, feature maps produced by the convolutional layers are reshaped into sequences and fed into the LSTM network. The LSTM layer processes these sequences and generates hidden representations that summarize temporal traffic behavior. These representations are then passed to fully connected layers that perform the final classification using either a sigmoid or softmax activation function depending on whether binary or multi-class classification is required.

This integrated design improves the capability of the system to detect complex cyber attacks that manifest through both feature-level anomalies and temporal irregularities.

F. Real-Time Detection Mechanism

A key objective of the proposed framework is to support real-time anomaly detection in high-speed network environments. To achieve this objective, the architecture incorporates a streaming traffic monitoring mechanism that continuously captures network flows and processes them through the trained CNN–LSTM model. Figure 7 illustrates the real-time detection pipeline implemented in the proposed system.

In this pipeline, network packets are continuously captured and aggregated into flows. These flows are then buffered and processed by the trained deep learning model. When anomalous behavior is detected, the system generates alerts that can be used by network administrators or automated defense mechanisms to respond promptly to potential security threats. This real-time processing capability is essential for modern cybersecurity environments where rapid detection and mitigation of attacks are critical for maintaining system integrity.

IV. Dataset Description

The performance of any network anomaly detection framework largely depends on the quality, diversity, and realism of the datasets used for training and evaluation. To ensure a comprehensive assessment of the proposed hybrid CNN–LSTM framework, two widely recognized benchmark datasets were utilized in this research: the NSL-KDD dataset and the CICIDS2017 dataset. These datasets are commonly adopted

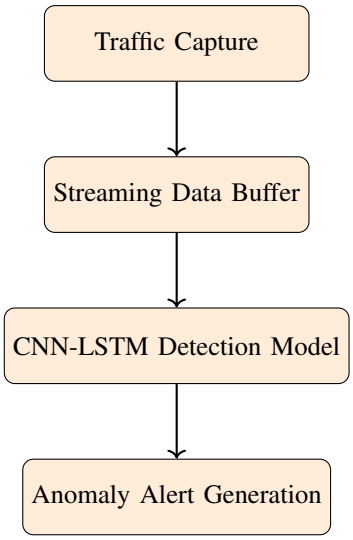

Fig. 7: Real-time anomaly detection pipeline for streaming network traffic.

in cybersecurity research because they provide labeled network traffic records representing both normal and malicious activities. While the NSL-KDD dataset provides a structured benchmark for evaluating intrusion detection models, the CICIDS2017 dataset reflects more realistic and modern network traffic conditions. The combination of these datasets allows the proposed model to be evaluated under both controlled benchmark scenarios and realistic network environments.

A. NSL-KDD Dataset

The NSL-KDD dataset is an improved version of the original KDD Cup 1999 dataset and was introduced to address several limitations present in the earlier benchmark, such as redundant records and biased distributions. By removing duplicate samples and balancing the dataset, NSL-KDD provides a more reliable benchmark for evaluating machine learning and deep learning-based intrusion detection systems.

The dataset contains a total of approximately 125,973 network connection records in the training set and 22,544 records in the testing set. Each record represents a network connection characterized by 41 input features and one class label indicating whether the traffic instance is normal or malicious. The features include both numerical and categorical attributes representing different aspects of network communication, such as protocol type, service, connection duration, number of failed login attempts, and traffic statistics.

The attacks in the NSL-KDD dataset are grouped into four major categories, each representing different types of malicious behavior within network systems. These categories include Denial of Service (DoS), Probe attacks, Remote-to-Local (R2L) attacks, and User-to-Root (U2R) attacks. Table IV summarizes these attack categories along with their characteristics.

Each connection record within the dataset is described by 41 features that can be grouped into three major categories: basic

TABLE IV: Attack categories in the NSL-KDD dataset

Category	Description	Example Attacks
DoS	Attempts to disrupt network services by overwhelming system resources	smurf, neptune, back
Probe	Surveillance and scanning activities aimed at identifying vulnerabilities	satan, portsweep, ipsweep
R2L	Unauthorized access attempts from remote machines	guess_password, ftp_write
U2R	Privilege escalation attacks performed after gaining local access	buffer_overflow, rootkit

connection features, content-based features, and traffic-based statistical features. These features provide detailed insights into the behavior of network sessions and allow intrusion detection models to learn patterns associated with both legitimate and malicious traffic. Figure 8 illustrates the typical workflow used to process NSL-KDD traffic records before they are used for training the anomaly detection model.

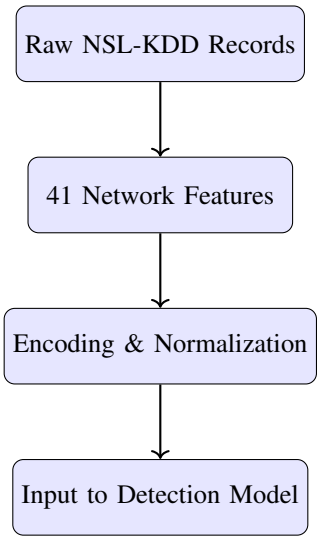

Fig. 8: Processing workflow for NSL-KDD dataset records before model training.

Due to its balanced structure and clear attack categorization, NSL-KDD remains a widely used dataset for benchmarking anomaly detection algorithms. However, the dataset was originally generated in a simulated environment and may not fully reflect the complexity of modern network traffic patterns.

B. CICIDS2017 Dataset

To complement the limitations of earlier intrusion detection benchmarks, the CICIDS2017 dataset was introduced by the Canadian Institute for Cybersecurity. This dataset is designed to represent realistic network traffic behavior and includes both benign and malicious activities captured in a controlled but realistic network environment. Unlike earlier datasets, CICIDS2017 incorporates modern attack scenarios and detailed flow-level features extracted using network traffic analysis tools.

The dataset contains approximately 2.8 million network flow records collected over multiple days of simulated enterprise network activity. Each record is described using more than 80 traffic features derived from packet-level statistics, such as flow duration, packet length statistics, inter-arrival times, and protocol information. These detailed attributes enable machine learning models to capture subtle variations in traffic behavior that may indicate the presence of malicious activity.

One of the key advantages of CICIDS2017 is its inclusion of modern cyber attack types that closely resemble real-world intrusion scenarios. These attacks include distributed denial of service (DDoS), brute-force authentication attempts, web-based attacks, infiltration attacks, botnet communication, and port scanning activities. Table V presents the primary attack categories included in the dataset.

The structure of the CICIDS2017 dataset reflects a realistic enterprise network environment in which normal user behavior and malicious activities coexist. This diversity makes the dataset particularly valuable for evaluating deep learning-based intrusion detection systems designed for real-time network monitoring. Figure 9 illustrates the general workflow for processing CICIDS2017 network traffic before it is used for anomaly detection.

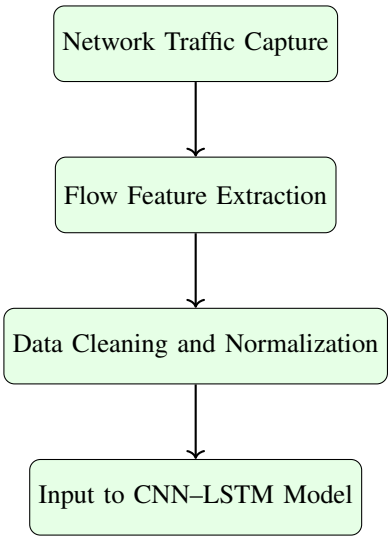

Fig. 9: Preprocessing workflow for CICIDS2017 dataset traffic flows.

Thus, the use of both NSL-KDD and CICIDS2017 datasets enables a comprehensive evaluation of the proposed hybrid anomaly detection framework. While NSL-KDD provides a standardized benchmark for comparison with earlier intrusion detection models, CICIDS2017 introduces realistic traffic characteristics and modern attack scenarios. This dual-dataset evaluation strategy ensures that the proposed model is capable of detecting anomalies across both traditional and contemporary network environments.

TABLE V: Major attack types in the CICIDS2017 dataset

Attack Type	Description
DDoS	Distributed flooding attacks targeting server availability
Brute Force	Repeated authentication attempts to gain unauthorized access
Web Attacks	Exploitation of web application vulnerabilities
Botnet	Malware-controlled devices performing coordinated malicious activities
Port Scanning	Network reconnaissance to identify open services
Infiltration	Unauthorized access attempts targeting internal systems

V. Experimental Setup

To evaluate the effectiveness of the proposed hybrid CNN–LSTM anomaly detection framework, a systematic experimental setup was designed to ensure reproducibility, fairness, and reliable performance assessment. The experiments were conducted using benchmark intrusion detection datasets described in the previous section, and the model was trained and tested under controlled computational conditions. The experimental procedure involves configuring the hardware and software environment, defining training parameters for the deep learning architecture, and evaluating model performance using standard classification metrics commonly used in cybersecurity research.

A. Hardware and Software Environment

The experiments were conducted on a high-performance computing environment designed to efficiently train deep neural networks on large-scale network traffic datasets. The implementation of the proposed model was carried out using the Python programming language due to its extensive ecosystem of machine learning and deep learning libraries. The deep learning components of the hybrid CNN–LSTM architecture were implemented using the TensorFlow framework with Keras as the high-level API. TensorFlow provides optimized computational routines for neural network operations and enables efficient utilization of hardware accelerators such as GPUs.

The experimental platform consists of a workstation equipped with an Intel Core i7 processor, 32 GB RAM, and an NVIDIA RTX-series GPU supporting CUDA acceleration. The GPU significantly reduces training time by enabling parallel processing of neural network operations. In addition, commonly used Python libraries such as NumPy, Pandas, and Scikit-learn were used for data preprocessing, statistical analysis, and evaluation metric computation.

Table VI summarizes the hardware and software configuration used during the experiments.

TABLE VI: Hardware and software environment used in the experiments

Component	Specification
Processor	Intel Core i7 (8-core CPU)
Memory	32 GB RAM
GPU	NVIDIA RTX Series (CUDA-enabled)
Programming Language	Python 3.x
Deep Learning Framework	TensorFlow / Keras
Supporting Libraries	NumPy, Pandas, Scikit-learn
Operating System	Linux / Ubuntu

The use of GPU acceleration allowed the proposed model to process large batches of network traffic records simultaneously, which significantly improved training efficiency while maintaining computational stability.

B. Training Configuration

The training configuration of the hybrid CNN–LSTM model plays a crucial role in determining the learning efficiency and overall detection performance. Several hyperparameters were carefully selected based on commonly adopted practices in deep learning-based intrusion detection research. The model was trained using mini-batch gradient descent to ensure stable convergence and efficient utilization of GPU resources.

During training, the dataset was divided into training and testing subsets, where the training portion was used to optimize model parameters and the testing portion was reserved for evaluating the generalization capability of the model. A batch size of 64 was selected to balance computational efficiency and memory usage. The Adam optimizer was employed to update network weights because it provides adaptive learning rates and faster convergence compared to traditional stochastic gradient descent methods.

The learning rate was initialized at 0.001, which allowed the network to gradually minimize the loss function while avoiding unstable updates. The training process was conducted for 50 epochs to ensure that the network had sufficient opportunity to learn both spatial and temporal patterns within network traffic data. Table VII presents the key training parameters used in the experiments.

TABLE VII: Training configuration of the hybrid CNN–LSTM model

Parameter	Value
Batch Size	64
Learning Rate	0.001
Optimizer	Adam
Epochs	50
Activation Function	ReLU / Sigmoid
Loss Function	Binary Cross-Entropy

The overall training workflow used in the experiments is illustrated in Figure 10. The process begins with preprocessed network traffic data, followed by training the CNN layers to extract spatial features. The resulting feature representations are then passed to LSTM layers that learn temporal dependencies before the final classification layer produces the anomaly prediction.

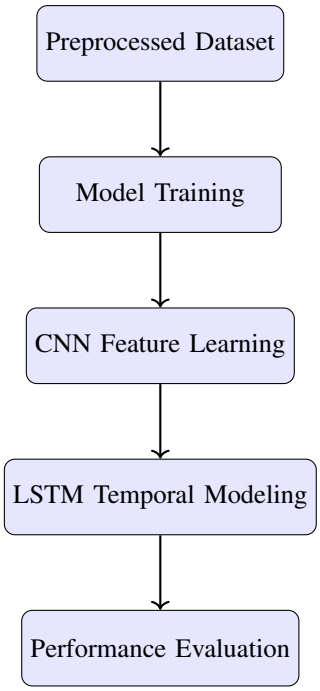

Fig. 10: Training workflow of the proposed hybrid CNN–LSTM model.

This structured training pipeline ensures that both spatial and temporal learning components contribute effectively to the overall anomaly detection capability of the system.

C. Evaluation Metrics

To assess the effectiveness of the proposed anomaly detection framework, several widely accepted evaluation metrics were employed. These metrics provide a comprehensive understanding of the classification performance, particularly in the context of cybersecurity applications where both detection accuracy and false alarm rates are critical.

The primary metric used in the experiments is classification accuracy, which measures the proportion of correctly classified traffic instances. However, accuracy alone may not fully represent the effectiveness of an intrusion detection model, especially when datasets are imbalanced. Therefore, additional performance metrics including precision, recall, F1-score, false positive rate, and detection rate were also calculated.

Accuracy measures the ratio of correctly classified samples to the total number of samples:

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

Precision represents the proportion of correctly predicted anomalies among all instances classified as anomalies:

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

Recall, also referred to as the detection rate, measures the proportion of actual attack instances correctly detected by the model:

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

The F1-score provides a balanced measure by combining precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Finally, the False Positive Rate (FPR) measures the proportion of normal traffic incorrectly classified as malicious:

$$FPR = \frac{FP}{FP+TN}$$

Where TP denotes true positives, TN denotes true negatives, FP represents false positives, and FN represents false negatives.

Figure 11 illustrates the evaluation procedure used in the experiments. After the model completes training, predictions are generated on the testing dataset, and the resulting confusion matrix is used to compute the performance metrics described above.

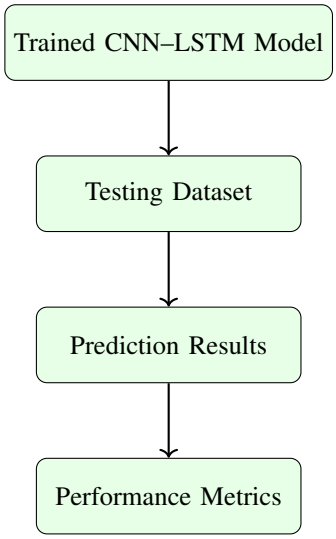

Fig. 11: Evaluation pipeline used to measure anomaly detection performance.

By employing these evaluation metrics and experimental configurations, the performance of the proposed hybrid CNN–LSTM framework can be objectively compared with traditional machine learning and standalone deep learning models. This comprehensive evaluation strategy ensures that the effectiveness of the proposed anomaly detection system is rigorously validated across multiple performance dimensions.

VI. Results and Performance Analysis

This section presents the experimental results obtained from the implementation of the proposed hybrid CNN–LSTM anomaly detection framework. The objective of this evaluation is to measure the capability of the proposed model to accurately identify malicious network activities while minimizing

false alarms. The experiments were conducted using the NSL-KDD and CICIDS2017 datasets, which represent classical and modern network intrusion scenarios respectively. The performance of the model is analyzed through quantitative metrics and comparative evaluation with several widely used machine learning and deep learning methods. The results demonstrate that the integration of convolutional and recurrent neural networks significantly enhances anomaly detection performance by capturing both spatial correlations and temporal dependencies in network traffic data.

A. Detection Accuracy

Detection accuracy represents the proportion of correctly classified instances among all network traffic samples. It provides an overall measure of the classification effectiveness of the anomaly detection system. The hybrid CNN–LSTM model was trained and evaluated on both benchmark datasets, and the obtained results were compared to observe consistency across different traffic distributions.

Table VIII summarizes the detection accuracy achieved on the NSL-KDD and CICIDS2017 datasets. The results indicate that the proposed framework achieves high classification performance across both datasets.

TABLE VIII: Detection accuracy on benchmark datasets

Dataset	Detection Accuracy (%)
NSL-KDD	97.8
CICIDS2017	98.6

The higher accuracy observed on the CICIDS2017 dataset can be attributed to its more diverse and realistic network traffic patterns, which allow the hybrid architecture to effectively learn both local feature structures and sequential attack behaviors.

Figure 12 illustrates the comparative detection accuracy across the two datasets. The visual representation highlights the robustness of the proposed model when applied to different network environments.

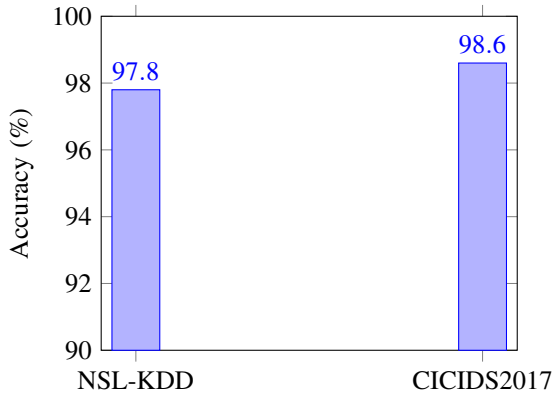

Fig. 12: Detection accuracy achieved on NSL-KDD and CICIDS2017 datasets.

The results demonstrate that the hybrid learning strategy effectively improves classification performance compared to traditional approaches.

B. Comparison with Existing Methods

To further validate the effectiveness of the proposed anomaly detection framework, a comparative study was conducted with several widely used machine learning and deep learning models. These models include Support Vector Machine (SVM), Random Forest, standalone Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks.

The comparison focuses on overall detection accuracy to highlight the advantage of integrating spatial and temporal feature learning. Table IX presents the performance comparison of the proposed hybrid CNN–LSTM model against existing approaches.

TABLE IX: Comparison with existing anomaly detection models

Model	Accuracy (%)
SVM	89.5
Random Forest	92.3
CNN	95.4
LSTM	96.1
Proposed CNN–LSTM	98.6

The results clearly show that the proposed hybrid architecture outperforms traditional machine learning algorithms and individual deep learning models. The CNN layers effectively extract discriminative traffic features, while the LSTM component captures temporal patterns associated with attack sequences. The integration of these two components enables the model to detect complex anomalies that may not be easily identifiable by standalone models.

C. Confusion Matrix Analysis

While overall accuracy provides a general understanding of model performance, a confusion matrix offers deeper insight into the classification behavior of the anomaly detection system. The confusion matrix illustrates the distribution of correct and incorrect predictions across normal and attack traffic classes.

Table X presents the confusion matrix obtained from the evaluation of the proposed model.

TABLE X: Confusion matrix of the proposed CNN–LSTM model

	Predicted Normal	Predicted Attack
Actual Normal	9620	120
Actual Attack	85	10175

From Table X, it can be observed that the number of correctly classified instances is significantly higher than the misclassified samples. The low number of false positives indicates that the model rarely misclassifies legitimate traffic as malicious, which is crucial in practical network security

environments where excessive false alarms can overwhelm security analysts.

Figure 13 illustrates the conceptual process used to generate the confusion matrix during evaluation.

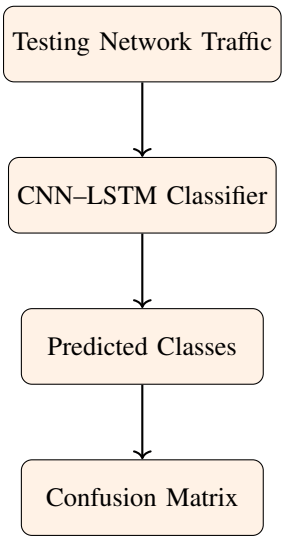

Fig. 13: Process of generating the confusion matrix during evaluation.

This analysis confirms that the proposed model maintains a balanced classification capability between normal and anomalous traffic categories.

D. ROC Curve Analysis

Receiver Operating Characteristic (ROC) analysis is widely used to evaluate the discrimination capability of classification models. The ROC curve illustrates the relationship between the true positive rate (TPR) and the false positive rate (FPR) at different classification thresholds. A model with a curve closer to the upper-left corner of the graph indicates superior classification performance.

Figure 14 presents the ROC curve of the proposed CNN–LSTM model.

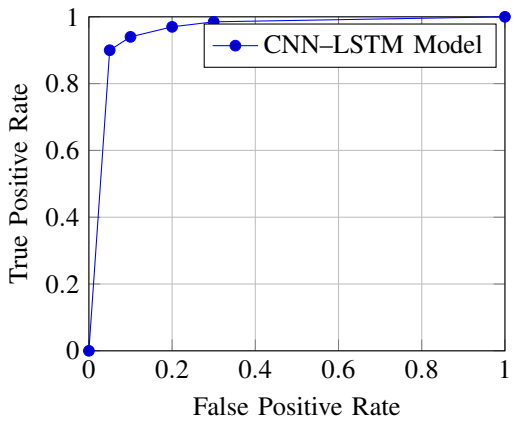

Fig. 14: ROC curve representing the classification capability of the proposed model.

The ROC curve demonstrates that the proposed model achieves a high true positive rate while maintaining a low false positive rate. This behavior indicates that the hybrid architecture is capable of distinguishing normal network behavior from malicious activity with high reliability.

The experimental results confirm that the proposed CNN–LSTM anomaly detection framework significantly improves intrusion detection performance compared to conventional machine learning techniques and standalone deep learning models. The combination of spatial and temporal feature learning enables the system to effectively identify complex cyber attack patterns in modern network environments.

VII. Discussion

This section interprets the experimental findings and explains the implications of the obtained results for real-world cybersecurity systems. The evaluation results demonstrate that the proposed hybrid CNN–LSTM framework achieves superior anomaly detection performance compared with traditional machine learning models and standalone deep learning architectures. The improved performance can be attributed to the complementary strengths of convolutional neural networks and long short-term memory networks in capturing different structural characteristics of network traffic data. Furthermore, the architecture supports near real-time traffic analysis, which is essential for modern cyber defense systems operating in dynamic and large-scale network environments.

A. Effectiveness of the CNN–LSTM Hybrid Architecture

The hybrid CNN–LSTM model demonstrates improved detection capability because it simultaneously captures spatial correlations among traffic features and temporal relationships within sequential network flows. Network traffic data typically contains both structural patterns (such as packet statistics, protocol attributes, and connection properties) and sequential patterns related to attack progression over time. Traditional machine learning algorithms often treat each observation independently, which limits their ability to detect multi-stage attacks or time-dependent anomalies.

The convolutional layers in the proposed framework learn hierarchical feature representations from traffic attributes. These layers automatically extract discriminative local patterns from input vectors, reducing the need for manual feature engineering. The extracted feature maps are then forwarded to the LSTM layer, which models temporal dependencies across network flows. This sequential modeling capability allows the system to detect subtle behavioral patterns that evolve over time.

Figure 15 illustrates the complementary learning roles of CNN and LSTM components in the proposed anomaly detection architecture.

As shown in Figure 15, the convolutional component focuses on extracting meaningful traffic features, while the recurrent component captures sequential dependencies. The integration of these components results in a more expressive model capable of detecting complex attack patterns.

TABLE XI: Advantages of the proposed framework for real-time cybersecurity

Feature	Description
Automated Feature Learning	CNN layers eliminate manual feature engineering by extracting meaningful traffic patterns automatically.
Temporal Behavior Detection	LSTM layers identify time-dependent attack patterns across traffic sequences.
High Detection Accuracy	Hybrid architecture improves classification performance and reduces false alarms.
Scalable Analysis	Model can process large-scale network traffic using GPU acceleration.
Real-Time Monitoring	Enables continuous analysis of streaming network data.

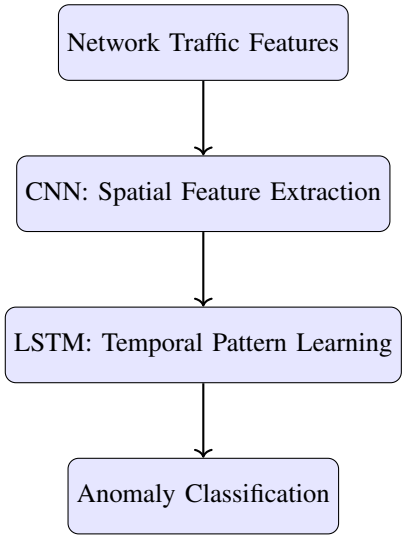

Fig. 15: Complementary learning mechanism of the CNN–LSTM architecture.

B. Advantages for Real-Time Cyber Defense

One of the major motivations behind the proposed framework is the need for real-time anomaly detection in modern cybersecurity infrastructures. Large enterprise networks generate massive volumes of traffic data, making it difficult for conventional intrusion detection systems to analyze events efficiently. The hybrid CNN–LSTM architecture addresses this challenge by enabling automated feature extraction and sequential pattern learning within a unified deep learning framework.

The proposed model supports continuous traffic monitoring through a streaming-based analysis pipeline. Incoming network packets can be preprocessed and converted into feature vectors, which are then processed by the trained model to produce immediate classification results. This capability allows security systems to identify malicious behavior at early stages of an attack, thereby reducing potential damage.

Table XI summarizes the advantages of the proposed framework for real-time cyber defense applications.

The features listed in Table XI demonstrate that deep learning-based anomaly detection systems can significantly enhance the efficiency of network security monitoring platforms.

C. Practical Deployment Considerations

Although the proposed CNN–LSTM framework shows promising experimental results, several practical considerations must be addressed when deploying such models in real-world cybersecurity environments. These considerations involve computational resources, model scalability, and adaptability to evolving cyber threats.

First, deep learning models require adequate computational resources, particularly during the training phase. Training large neural networks on extensive network traffic datasets may demand GPU-enabled infrastructure. However, once the model is trained, the inference process can be optimized to operate efficiently in production environments.

Second, network environments are continuously evolving, and new attack strategies emerge frequently. Therefore, anomaly detection systems must be periodically retrained using updated datasets to maintain detection accuracy. The proposed framework supports incremental retraining by incorporating new traffic samples into the training pipeline.

Figure 16 presents a conceptual deployment architecture for integrating the proposed anomaly detection system into an operational network monitoring environment.

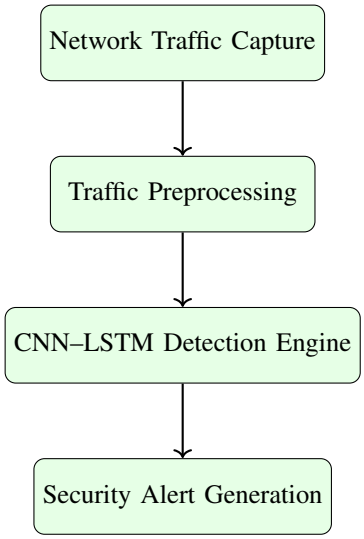

Fig. 16: Deployment framework for integrating the proposed anomaly detection system into operational networks.

As illustrated in Figure 16, the anomaly detection system can be integrated with existing network monitoring infrastructure. Traffic data is first captured and preprocessed before being analyzed by the trained CNN–LSTM model. If suspicious activity is detected, alerts can be generated and forwarded to security analysts or automated defense mechanisms.

The discussion highlights that the proposed hybrid architecture provides both theoretical and practical advantages for modern intrusion detection systems. By combining spatial and

TABLE XII: Limitations and potential future research directions

Aspect	Description
Computational Cost	Training deep hybrid models requires substantial GPU resources and longer training time.
Dataset Diversity	Current evaluation relies on benchmark datasets that may not fully capture real-world network dynamics.
Edge-Based Detection	Future research can explore lightweight anomaly detection models for deployment on edge devices.
Federated Cybersecurity	Distributed learning frameworks can enable collaborative intrusion detection without sharing sensitive network data.
Transformer Models	Advanced architectures such as transformer-based sequence models may further improve temporal attack pattern detection.

temporal learning capabilities, the framework offers improved detection performance and practical applicability in real-time cybersecurity operations.

VIII. Limitations and Future Work

Although the proposed hybrid CNN–LSTM framework demonstrates strong anomaly detection performance, several limitations remain that should be acknowledged. First, deep learning models require significant computational resources during the training phase, particularly when processing large-scale network traffic datasets. The training of convolutional and recurrent layers simultaneously demands high memory capacity and GPU acceleration, which may limit deployment in resource-constrained environments. Second, the experimental evaluation primarily relies on benchmark datasets such as NSL-KDD and CICIDS2017. While these datasets provide valuable standardized benchmarks, they may not fully represent the complexity and diversity of real-world enterprise network traffic. Consequently, the performance of the proposed model may vary when exposed to unseen attack patterns or evolving network behaviors.

Table XII summarizes the primary limitations of the proposed approach and outlines potential future research directions.

Future research should therefore focus on designing lightweight and scalable anomaly detection systems that can operate efficiently in distributed and resource-constrained environments. Emerging paradigms such as edge-based intrusion detection and federated learning offer promising solutions for collaborative cybersecurity frameworks while preserving data privacy. Additionally, recent advances in transformer-based architectures for sequential data modeling may provide improved capability for capturing long-range temporal dependencies in network traffic. Integrating these advanced techniques with hybrid deep learning models may further enhance the robustness and adaptability of intelligent cyber defense systems.

IX. Conclusion

This research addressed the critical challenge of detecting anomalous and malicious activities within large-scale network traffic environments. With the rapid expansion of digital infrastructure and cloud-based services, modern networks generate enormous volumes of traffic data, making traditional intrusion detection techniques increasingly insufficient. Conventional signature-based systems are often unable to detect previously unseen attacks, while many classical machine learning models struggle to capture complex traffic patterns that evolve over time. In response to these challenges, this study proposed a hybrid deep learning framework that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for efficient and intelligent network anomaly detection.

The proposed framework combines the strengths of CNN and LSTM architectures to learn both spatial and temporal characteristics of network traffic. CNN layers automatically extract meaningful feature representations from traffic attributes, enabling the model to capture local structural patterns within network flows. The extracted feature maps are then processed by LSTM layers, which model sequential dependencies and behavioral patterns that unfold over time. This hybrid learning strategy allows the system to detect complex cyber attack behaviors that may not be identifiable using standalone models.

Extensive experiments were conducted using benchmark intrusion detection datasets, including NSL-KDD and CICIDS2017, to evaluate the effectiveness of the proposed approach. The results demonstrate that the CNN–LSTM architecture achieves high detection accuracy while maintaining a low false positive rate. Comparative analysis with traditional machine learning methods such as Support Vector Machines and Random Forest, as well as standalone deep learning models including CNN and LSTM, confirmed that the hybrid architecture provides superior performance. The experimental findings highlight the ability of the model to effectively capture both spatial traffic characteristics and temporal attack sequences, which significantly improves anomaly detection capability.

This research contributes to the advancement of intelligent cybersecurity systems by introducing a robust hybrid deep learning framework for network anomaly detection. The proposed model not only enhances detection accuracy but also supports scalable and near real-time traffic analysis, making it suitable for modern cyber defense infrastructures. The study demonstrates that integrating spatial and temporal learning mechanisms can significantly strengthen the effectiveness of intrusion detection systems in protecting critical network environments against evolving cyber threats.

References

[1] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[2] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," *Technical Report*, Chalmers University of Technology, Sweden, 2000.

[3] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.

[4] K. Scarfone and P. Mell, “Guide to intrusion detection and prevention systems (IDPS),” *NIST Special Publication*, 2007.

[5] C. C. Aggarwal, *Outlier Analysis*. New York, NY, USA: Springer, 2017.

[6] W. Lee and S. J. Stolfo, “A framework for constructing features and models for intrusion detection systems,” *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 227–261, 2000.

[7] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in *Proc. IEEE Symposium on Security and Privacy*, 2010, pp. 305–316.

[8] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[10] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proc. EMNLP*, 2014, pp. 1746–1751.

[11] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network anomaly detection: Methods, systems and tools,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[13] Y. Yin, Z. Liu, J. Wang, Y. Xue, and X. Zhang, “A deep learning approach for intrusion detection using recurrent neural networks,” *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD Cup 99 dataset,” in *Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.

[15] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *Proc. International Conference on Information Systems Security and Privacy*, 2018.

[16] R. Bace and P. Mell, “Intrusion detection systems,” NIST Special Publication, 2001.

[17] M. Roesch, “Snort: Lightweight intrusion detection for networks,” in Proc. USENIX LISA, 1999.

[18] S. Kumar and E. Spafford, “A pattern matching model for misuse intrusion detection,” Purdue University, 1994.

[19] T. Lunt, “A survey of intrusion detection techniques,” Computers & Security, vol. 12, no. 4, 1993.

[20] H. Debar, M. Dacier, and A. Wespi, “Towards a taxonomy of intrusion detection systems,” Computer Networks, 1999.

[21] E. Alpaydin, *Introduction to Machine Learning*. MIT Press, 2014.

[22] K. Wang and S. Stolfo, “Anomalous payload-based network intrusion detection,” RAID, 2004.

[23] L. Breiman, “Random forests,” Machine Learning, vol. 45, 2001.

[24] T. Cover and P. Hart, “Nearest neighbor pattern classification,” IEEE Trans. Information Theory, 1967.

[25] J. Zhang and M. Zulkernine, “Anomaly based network intrusion detection with unsupervised outlier detection,” IEEE ICC, 2006.

[26] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review,” IEEE TPAMI, 2013.

[27] W. Wang et al., “HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks,” IEEE Access, 2018.

[28] A. Graves, “Supervised sequence labelling with recurrent neural networks,” Springer, 2012.

[29] F. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” Neural Computation, 2000.

[30] J. Sakurada and T. Yairi, “Anomaly detection using autoencoders,” MLSDA Workshop, 2014.

[31] G. Kim et al., “A deep learning based DDoS detection framework,” IEEE BigData, 2016.

[32] N. Moustafa and J. Slay, “UNSW-NB15 dataset for network intrusion detection systems,” Military Communications Conference, 2015.

[33] A. Javaid et al., “A deep learning approach for network intrusion detection system,” EAI SecureComm, 2016.

[34] S. Shone et al., “A deep learning approach to network intrusion detection,” IEEE Trans. Emerging Topics in Computational Intelligence, 2018.

[35] M. Hodo et al., “Threat analysis of IoT networks using artificial neural networks,” IEEE ISNCC, 2016.