International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 2, 2025

Peer Reviewed, Referred & Indexed Journal

Research Paper

A Hybrid AI-Driven Cybersecurity Framework for Secure Communication Systems in ECE Networks

Dr. Anumolu Lasmika[1] and Dr. Pradeep G[2]
Academic Consultants, Department of Electronics and Communication Engineering,
Sri Venkateswara University College of Engineering, Sri Venkateswara University Tirupati, Andhra Pradesh
Anumolu.lasmika@gmail.com[1]
Associate Professor, Department of Computer Science and Engineering,
Chadalawada Ramanamma Engineering College ,Tirupati, Andhrapradesh
gpc.tpt@gmail.com[2]

Abstract: The rapid growth of interconnected communication systems in Electronic and Communication Engineering (ECE) networks has increased the risk of cyber threats, making secure data transmission a critical challenge. Traditional intrusion detection approaches often struggle to handle dynamic attack patterns and complex network behaviour, resulting in limited accuracy and high false alarm rates. The objective of this study is to develop an efficient and adaptive cybersecurity framework capable of improving intrusion detection performance in modern communication environments.The proposed work introduces a hybrid AI-driven cybersecurity framework that integrates multiple learning techniques to analyse network traffic effectively. The system utilizes a structured pipeline involving data preprocessing, feature engineering, and hybrid model design combining spatial and temporal learning mechanisms. The KDD Cup 1999 dataset is used for training and evaluation, with features categorized into basic, content-based, time-based, and host-based attributes. The model is trained using optimized parameters and evaluated through standard performance metrics to ensure reliability and reproducibility.Experimental results demonstrate that the proposed framework achieves an accuracy of 95.2%, precision of 94.1%, recall of 93.6%, and an AUC score of 0.95, outperforming traditional machine learning and standalone deep learning models. The results also indicate stable convergence, reduced loss, and improved detection capability across multiple attack categories.The study concludes that the hybrid approach significantly enhances intrusion detection performance while supporting real-time adaptability and scalability. The framework provides a practical solution for securing ECE

communication systems and has strong potential for deployment in real-world applications such as IoT, smart grids, and industrial networks

KeywordsCybersecurity, Intrusion Detection System, Hybrid AI, ECE Networks, Deep Learning, Network Security, KDD Dataset, Threat Detection.

Received: 02-05-2025 Accepted: 11-06-2025 Published: 18-06-2025

1. Introduction

1.1 Background and Motivation

With the rapid growth of communication technologies and interconnected electronic systems, the demand for secure and reliable communication networks has increased significantly. Modern Electronic and Communication Engineering (ECE) networks form the backbone of critical infrastructures such as smart grids, industrial automation, healthcare systems, and IoT-based environments. As these systems become more interconnected, they also become more vulnerable to cyber threats, including intrusion attacks, denial-of-service attacks, and data breaches. Ensuring the security of such communication systems has become a major concern for both researchers and industry professionals.

Traditional cybersecurity mechanisms were mainly designed for static environments and predefined attack patterns. However, the current network landscape is highly dynamic, with evolving attack strategies that are increasingly sophisticated and difficult to detect. Intrusion Detection Systems (IDS) play an important role in identifying malicious activities within network traffic. Early IDS models were largely based on signature-based or rule-based techniques, which were effective only for known attacks but failed to detect unknown or zero-day threats [3], [12]. As a result, there has been a shift towards intelligent and adaptive security mechanisms that can handle complex and large-scale network data.

Machine learning and deep learning techniques have gained attention in recent years due to their ability to learn patterns from large datasets and detect anomalies in network traffic [1], [5]. These approaches have shown promising results in improving detection accuracy and reducing false alarm rates. However, despite these advancements, several challenges still exist in designing efficient and scalable cybersecurity frameworks suitable for ECE communication systems.

1.2 Challenges in Existing Cybersecurity Approaches

Although significant progress has been made in the development of intrusion detection systems, existing solutions face multiple limitations that restrict their effectiveness in real-world scenarios. One of the major challenges is the dependency on outdated or imbalanced

datasets. Many studies rely heavily on benchmark datasets such as KDD Cup 1999, which do not fully represent modern network traffic patterns and emerging cyber threats [4], [16]. This limitation affects the generalization capability of models and reduces their performance when deployed in real-time environments.

Another key issue is the inability of traditional machine learning models to handle high-dimensional and complex network data efficiently. While techniques such as Support Vector Machines and Random Forests have been widely used, they often struggle with scalability and feature selection in large datasets [11], [14]. Deep learning approaches, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have improved performance by capturing spatial and temporal features of network traffic [9], [15]. However, these models require large computational resources and are prone to overfitting if not properly optimized.

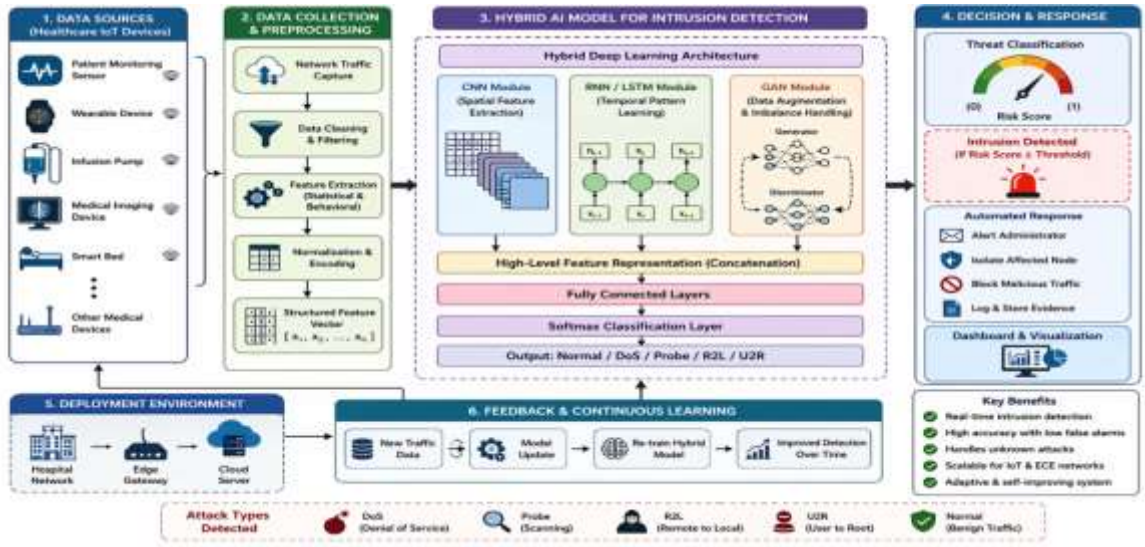

Fig 1:Hybrid AI-Based Cybersecurity Framework

The figure 1 shows a structured flow of a cybersecurity system designed for secure communication in ECE networks. It can be understood that the process begins with data collection from multiple connected devices, where network traffic is continuously monitored. The collected data is then cleaned and processed before being passed through different analytical layers. It is observed that a combination of learning techniques is used to capture both pattern-based and time-based behaviours in the data, which helps in identifying any unusual activity. Once a threat is detected, the system is able to classify it and take necessary action such as blocking suspicious traffic or alerting the administrator. It is also indicated that the system keeps improving over time by learning from new data, which makes it more reliable for real-time applications.

In addition to this, most existing frameworks focus on single-layer security mechanisms, which are insufficient for protecting multi-layer communication systems. For example, IoT and cyber-physical systems involve multiple layers, including device, network, and application levels, each of which requires dedicated security measures [6], [7]. The lack of integration between these layers creates vulnerabilities that attackers can exploit.

Another important challenge is the limited ability of current systems to adapt to evolving attack patterns. Reinforcement learning-based approaches have been proposed to address this issue, but they often suffer from slow convergence and high training complexity [8]. Similarly, anomaly detection techniques can identify unknown attacks but tend to produce high false positive rates, which affects system reliability [12].

Furthermore, the absence of real-time processing capabilities remains a significant drawback in many existing intrusion detection systems. In practical applications, cybersecurity frameworks must be capable of detecting and responding to threats instantly. However, many research models are evaluated only in offline environments, making them unsuitable for deployment in real-time communication systems [20], [23].

1.3 Proposed Approach and Research Contribution

To address the limitations of existing approaches, this study proposes a hybrid AI-driven cybersecurity framework designed specifically for secure communication systems in ECE networks. The proposed framework integrates multiple artificial intelligence techniques to improve detection accuracy, adaptability, and efficiency. By combining the strengths of different models, the framework aims to overcome the shortcomings of individual approaches and provide a robust solution for modern cybersecurity challenges.

The proposed system adopts a multi-layer architecture that incorporates both feature-based and behavior-based analysis of network traffic. It utilizes deep learning techniques such as CNN and RNN for extracting spatial and temporal features, respectively, while also incorporating generative models to enhance data diversity and handle class imbalance issues [19]. Additionally, the framework is designed to support scalable and efficient processing, making it suitable for real-time deployment in communication networks.

The use of hybrid models allows the system to achieve better performance compared to traditional methods. By integrating multiple techniques, the framework can effectively detect both known and unknown attacks, reduce false positives, and improve overall system reliability. Moreover, the proposed approach focuses on optimizing computational efficiency,

ensuring that the system can operate in resource-constrained environments such as IoT and embedded systems.

Another important aspect of the proposed framework is its adaptability to different network environments. Unlike conventional models that rely on fixed patterns, the hybrid approach enables continuous learning and adaptation, allowing the system to respond to emerging threats dynamically. This makes it highly suitable for modern ECE applications, where network conditions and attack patterns change frequently.

1.4 Key Contributions

The main contributions of this research work are summarized as follows:

- **Development of a Hybrid AI-Based Cybersecurity Framework:**A novel framework is proposed that integrates multiple artificial intelligence techniques, including deep learning and generative models, to enhance intrusion detection performance in communication networks.
- **Improved Detection Accuracy and Efficiency:**The proposed system achieves better accuracy and reduced false positive rates by combining spatial, temporal, and behavioral analysis of network traffic.
- **Scalable and Adaptive Security Solution for ECE Networks:**The framework is designed to support real-time processing and adaptability, making it suitable for modern communication systems, including IoT and cyber-physical environments.

The remainder of this paper is organized as follows. Section 2 presents a detailed review of existing literature related to cybersecurity and intrusion detection systems. Section 3 describes the dataset, preprocessing techniques, and the proposed methodology. Section 4 discusses the experimental aetup and performance evaluation of the proposed framework. Section 5 provides the experimental resultsand comprehensive discussion of the findings, and finally, Section 6 concludes the paper with future research directions.

2. Related Work / Literature Review

2.1 Machine Learning-Based Intrusion Detection Systems

Early research in intrusion detection mainly focused on machine learning techniques to identify malicious activities in network traffic. It is observed that traditional models such as decision trees, support vector machines, and random forests were widely used due to their simplicity and ease of implementation. Studies have shown that these approaches are effective in detecting known attack patterns but often struggle when dealing with unknown or

evolving threats [1], [11], [14]. It can be understood that these models depend heavily on feature engineering and require careful selection of input parameters to achieve acceptable performance.

Another important concern highlighted in earlier works is the assumption of a closed and static environment. It has been pointed out that real-world networks are highly dynamic, and machine learning models trained on static datasets may not perform well in practical scenarios [2]. This limitation reduces the adaptability of such systems and makes them less reliable for real-time deployment. Although these methods provide a strong foundation, they are not sufficient to handle the increasing complexity of modern communication networks.

2.2 Anomaly-Based and Signature-Based Approaches

Intrusion detection techniques are generally categorized into anomaly-based and signature-based methods. Signature-based systems rely on predefined patterns to detect attacks, which makes them efficient for identifying known threats. However, they fail to detect new or unknown attacks, which is a major drawback in modern cybersecurity environments [3]. On the other hand, anomaly-based systems attempt to identify deviations from normal behaviour, allowing them to detect previously unseen attacks.

While anomaly-based techniques offer better flexibility, they often suffer from high false alarm rates, which affects their usability in real-world applications [12]. It is also noted that maintaining a balance between detection accuracy and false positive rate remains a challenging task. Many studies have attempted to combine both approaches to improve performance, but the lack of proper integration strategies limits their effectiveness. Therefore, there is a need for more advanced and adaptive solutions that can address these issues.

2.3 Deep Learning-Based Intrusion Detection

With the advancement of computational power, deep learning techniques have been increasingly applied in cybersecurity. Models such as convolutional neural networks and recurrent neural networks have shown promising results in capturing complex patterns in network traffic data [5], [9], [15]. It is observed that CNN models are effective in extracting spatial features, while RNN-based models are useful for analysing sequential data.

Despite their advantages, deep learning models introduce new challenges. One major issue is the requirement for large amounts of training data, which is not always available in real-world scenarios. In addition, these models are computationally intensive and may not be suitable for resource-constrained environments such as IoT devices [6]. Another limitation is

the risk of overfitting, especially when the dataset is imbalanced or lacks diversity. These factors highlight the need for hybrid approaches that can combine the strengths of different models while minimizing their weaknesses.

2.4 Dataset Challenges in Cybersecurity Research

Datasets play a crucial role in the development and evaluation of intrusion detection systems. It has been observed that many studies rely on standard datasets such as KDD Cup 1999 and NSL-KDD, which are widely used but have several limitations [4]. These datasets are often outdated and do not reflect current network traffic patterns or modern attack strategies. As a result, models trained on these datasets may not generalize well to real-world environments.

Recent research has focused on developing new datasets such as UNSW-NB15 and CICIDS to address these issues [16], [17]. These datasets provide more realistic traffic patterns and include a wider range of attack types. However, even these datasets have limitations in terms of scalability and diversity. It is also noted that the lack of standardized evaluation metrics makes it difficult to compare different models effectively. Therefore, improving dataset quality and evaluation methods remains an important research area.

2.5 Advanced AI Techniques and Hybrid Models

To overcome the limitations of individual models, researchers have explored hybrid approaches that combine multiple techniques. It is understood that integrating machine learning and deep learning methods can improve detection accuracy and reduce false positives. For example, combining CNN and RNN models allows the system to capture both spatial and temporal features of network traffic [5].

Generative models such as GANs have also been introduced to address data imbalance issues by generating synthetic samples [19]. Similarly, autoencoder-based models have been used for anomaly detection, offering efficient representation of normal behaviour [20]. Reinforcement learning techniques have been applied to develop adaptive systems that can learn from dynamic environments, although they require careful tuning and high computational resources [8].

In the context of IoT and communication networks, hybrid models are particularly useful due to the complex and distributed nature of these systems [6], [7]. However, most existing works still focus on specific components rather than providing a unified framework. This creates a gap in the development of comprehensive cybersecurity solutions.

2.6 Application-Specific Cybersecurity Systems

Recent studies have focused on applying intrusion detection techniques to specific domains such as IoT, smart grids, and edge computing environments. It is observed that these applications introduce additional challenges due to resource constraints and heterogeneous network structures. For instance, IoT-based systems require lightweight and energy-efficient security mechanisms [21], [22].

In smart grid environments, ensuring secure communication is critical for maintaining system stability and reliability [24]. Similarly, edge computing systems require distributed security solutions that can operate efficiently across multiple nodes [25]. Although these studies provide valuable insights, they often address only specific aspects of cybersecurity and lack a comprehensive approach that integrates multiple layers of protection.

2.7 Research Gaps and Motivation

From the above analysis, several research gaps can be identified. First, most existing models lack the ability to handle real-time and dynamic network environments effectively. Second, there is a heavy reliance on outdated datasets, which limits the generalization capability of the models. Third, current approaches do not fully utilize the potential of hybrid AI techniques to improve detection performance.

In addition, there is a need for scalable and adaptive frameworks that can be applied to modern ECE communication systems. Existing solutions often focus on individual components rather than providing an integrated approach. This study aims to address these gaps by proposing a hybrid AI-driven cybersecurity framework that combines multiple techniques to achieve improved accuracy, efficiency, and adaptability.

Table 1: Comparison of Existing Approaches

Ref	Method	Accuracy	Efficiency	Challenges
[1]	ML-based IDS	Moderate	High	Limited adaptability
[3]	Signature + Anomaly IDS	Moderate	Medium	Cannot detect unknown attacks
[5]	Deep Learning IDS	High	Low	High computation cost
[6]	IoT Security ML/DL	Moderate	Medium	Scalability issues
[9]	Deep Neural IDS	High	Low	Overfitting
[11]	Random Forest	Moderate	High	Feature dependency

[15]	RNN-based IDS	High	Medium	Sequence complexity
[20]	Autoencoder IDS	High	Medium	Resource overhead
[22]	IoT Botnet Detection	High	Medium	Real-time limitations
[25]	Edge Security Model	Moderate	High	Distributed challenges

3. Methodology

3.1 Overall Framework Design

The proposed system is designed as a hybrid cybersecurity framework to ensure secure communication in ECE networks. It can be understood that the framework follows a structured pipeline starting from data collection to final threat detection and response. Network traffic data is continuously captured from communication channels and passed through multiple processing stages. Each stage is responsible for transforming raw data into a meaningful format that can be analysed effectively. The idea is to build a system that not only detects intrusions but also adapts to changing network behaviour over time.

It is observed that a multi-layer architecture has been adopted to improve detection capability. Instead of relying on a single technique, the framework combines different analytical approaches to capture various characteristics of network traffic. This layered design helps in identifying both simple and complex attack patterns. The integration of multiple components ensures better performance and reliability, especially in dynamic environments such as IoT and communication networks.

The proposed cybersecurity framework is designed to secure ECE communication networks through a hybrid learning pipeline that integrates statistical preprocessing, feature refinement, representation learning, and real-time threat response. Let $X \in \mathbb{R}^{n \times d}$ denote the raw network traffic matrix, where n is the number of traffic instances and d is the number of extracted attributes. The primary objective is to learn a mapping that separates benign and malicious traffic with high sensitivity while preserving computational efficiency.

The framework operates in four stages: data normalization, feature selection, hybrid model inference, and adaptive response. Each stage contributes to the overall reduction of uncertainty in the classification process. To formalize this setting, the observed network sample can be written as

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}, \quad (1)$$

where x_i represents the i-th traffic record and $y_i \in \{0,1,\ldots,C-1\}$ denotes the associated class label.

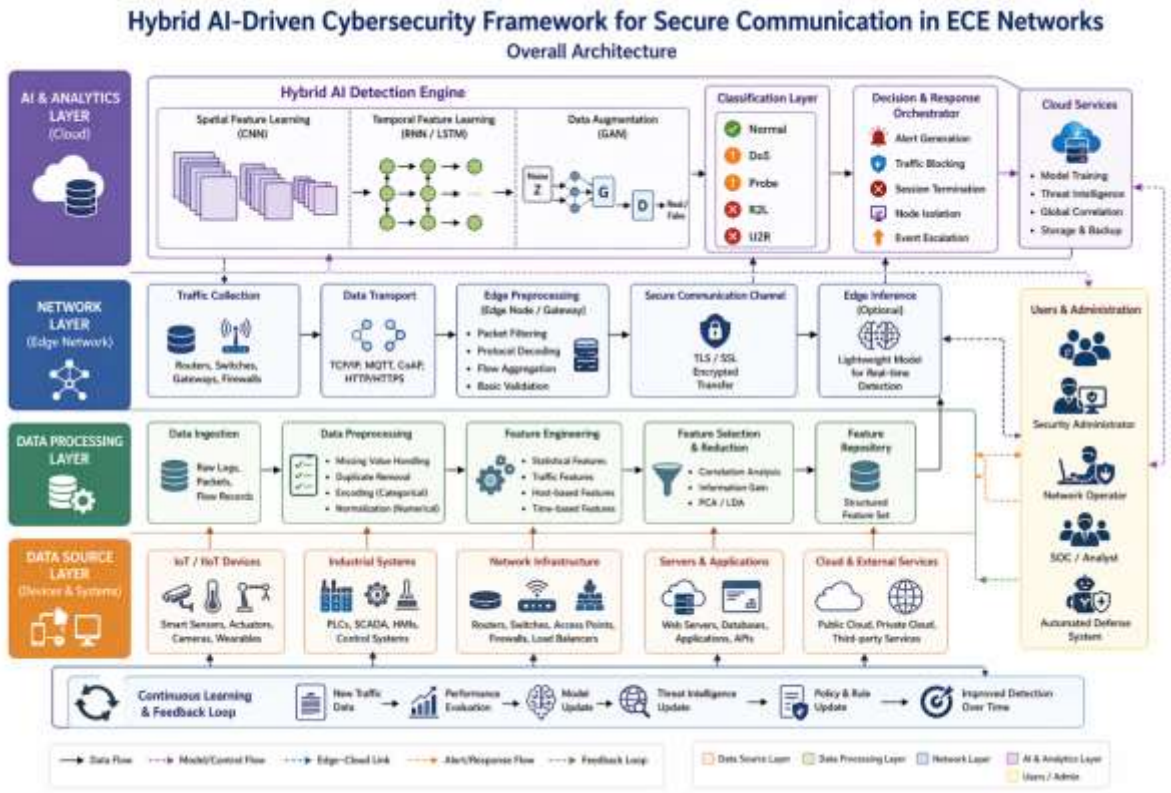

Fig 2:Multi-Layer Hybrid Cybersecurity Architecture

The figure shows a layered architecture designed for securing communication systems in ECE networks. It can be understood that the process starts from different data sources such as IoT devices, industrial systems, and network infrastructure, where communication data is continuously generated. This data is then passed through processing stages where it is cleaned, organized, and converted into a suitable format for further analysis. It is observed that the system applies multiple analytical steps, including feature extraction and selection, to focus on important patterns in the traffic. Further, the processed data is analysed in the AI layer, where different learning techniques are used together to identify any abnormal behaviour. Based on the results, the system is able to classify the activity and take necessary actions such as generating alerts or blocking suspicious traffic. It is also indicated that the system keeps updating itself through continuous feedback, which helps in handling new types of attacks.

3.2 Dataset Description and Preprocessing

The study makes use of a standard intrusion detection dataset, which contains network traffic records labelled as normal and malicious activities. Each record consists of multiple features

representing different aspects of a network connection, such as protocol type, data transfer size, and connection behaviour. It can be understood that these features play an important role in identifying patterns related to cyber-attacks. However, the raw dataset contains redundant and inconsistent values, which need to be handled before applying any learning technique.

To improve data quality, several preprocessing steps are carried out. Initially, duplicate records are removed to avoid bias in model training. Missing values, if any, are handled carefully to maintain data consistency. Categorical features are converted into numerical form using encoding techniques, while numerical attributes are normalized to ensure uniform scaling. These steps help in improving the performance of the model and reduce the chances of incorrect predictions.

In addition to this, the dataset includes a well-defined set of parameters that capture different characteristics of network traffic, which are essential for intrusion detection [26]. These parameters are broadly categorized into basic features, content-based features, time-based traffic features, and host-based features. Each category provides unique information about network behaviour, enabling the system to detect both simple and complex attack patterns. Understanding these parameters is important for effective feature selection and model training.

Table 2: Dataset Parameters Description [26]

Category	Parameters	Description
Basic Features	duration, protocol_type, service, flag, src_bytes, dst_bytes	Describe basic connection details and data transfer
Content Features	num_failed_logins, logged_in, root_shell, num_compromised	Capture suspicious login and access behaviour
Time-Based Traffic Features	count, srv_count, serror_rate, rerror_rate	Represent traffic patterns over short time intervals
Host-Based Features	dst_host_count, dst_host_srv_count, dst_host_same_srv_rate	Analyse long-term connection behaviour
Target Variable	label	Indicates normal or attack type (DoS, Probe, R2L, U2R)

The collected traffic records are first cleaned to remove redundancy, missing values, and scale inconsistency. Since network attributes often vary widely in magnitude, normalization is required to prevent dominance of high-variance features during training. For a given feature value x_{ij}, min-max normalization is defined as

$$\tilde{x}_{ij} = \frac{x_{ij} - x_j^{\min}}{x_j^{\max} - x_j^{\min} + \epsilon}, \quad (2)$$

where$x_j^{\min}$ and $x_j^{\max}$ are the minimum and maximum values of the j-th feature, and ϵ is a small constant for numerical stability.

For categorical variables such as protocol type or service type, one-hot encoding is applied. If a categorical attribute takes m possible states, its encoded representation is

$$\phi(c_k) = \begin{cases} 1, & \text{if } c = c_k, \\ 0, & \text{otherwise,} \end{cases} \quad k = 1,2,\dots,m. \quad (3)$$

After preprocessing, the refined feature vector is represented as

$$z_i = [\tilde{x}_{i1}, \tilde{x}_{i2}, \dots, \tilde{x}_{id'}]^\top, \quad (4)$$

whered' denotes the dimension after encoding and cleaning.

Algorithm 1: Dataset Preprocessing and Normalization

Algorithm 1 describes the preprocessing stage, where raw network traffic records are cleaned and transformed into a machine-readable form before training begins. In practical intrusion detection, raw data often contains duplicate rows, missing values, inconsistent labels, and mixed feature types such as numerical fields, categorical protocols, and service codes. For example, consider a synthetic record such as $[\text{duration} = 12, \text{protocol} = \text{tcp}, \text{service} = \text{http}, \text{srcbytes} = 430, \text{dstbytes} = 120]$. If another record is repeated or contains a missing destination byte value, the algorithm removes the duplicate and handles the missing entry through imputation or exclusion depending on the selected policy. Then categorical values like “tcp” and “http” are encoded into numeric vectors, and numerical attributes are normalized so that large-valued fields do not dominate smaller ones. This step is important because a model trained on unprocessed data may learn biased patterns and produce unstable predictions. By the end of this stage, the dataset becomes consistent, scaled, and ready for feature extraction and model learning.

Algorithm 1 Dataset Cleaning and Feature Normalization

1: **Input:** Raw traffic dataset $\mathcal{D}$, missing-value policy, feature list F

2: **Output:** Processed dataset $\mathcal{D}_p$

3: Remove duplicate records from $\mathcal{D}$

4: Detect missing values in each feature column

5: Impute or discard incomplete samples according to the selected policy

6: Encode categorical variables using one-hot or ordinal representation

7: Separate numerical and categorical attributes

8: Apply min-max normalization to each numerical feature

9: Standardize feature scale to reduce dominance of large-magnitude attributes

10: Reconstruct the cleaned feature matrix $\mathcal{D}_p$

11: Verify data consistency and label integrity

12: **Return:**$\mathcal{D}_p$

3.3 Feature Engineering and Selection

Feature engineering is an important step in improving the effectiveness of intrusion detection systems. It is observed that not all features contribute equally to the detection process. Some attributes may have little impact, while others play a major role in identifying attack patterns. Therefore, selecting the most relevant features helps in reducing computational complexity and improving model accuracy.

In this work, statistical and correlation-based techniques are used to identify important features from the dataset. Dimensionality reduction methods such as principal component analysis can also be applied to remove redundant information. By focusing only on significant attributes, the system becomes more efficient and faster in processing data. This step is particularly useful when dealing with large datasets, where unnecessary features can slow down the overall performance.

Not all attributes contribute equally to intrusion detection. Therefore, feature relevance is estimated using variance, correlation, and discriminative contribution. Let f_j denote the j-th candidate feature and let y denote the class variable. The correlation coefficient between a feature and the class label is given by

$$\rho(f_j, y) = \frac{\mathrm{cov}(f_j, y)}{\sigma_{f_j}\sigma_y}, \quad (5)$$

where$\mathrm{cov}(\cdot,\cdot)$ is covariance and σ denotes standard deviation.

To quantify the overall importance of a feature subset S, an objective function can be defined as

$$J(S) = \alpha \sum_{f_j \in S} \left|\rho(f_j, y)\right| - \beta \sum_{f_p, f_q \in S} \left|\rho(f_p, f_q)\right|, \quad (6)$$

where α and β are weighting parameters that balance class relevance and inter-feature redundancy.

The selected feature subset is then expressed as

3.4 Hybrid AI Model Design

The proposed framework uses a hybrid approach by combining multiple learning techniques to enhance detection performance. It can be understood that different models are suitable for different types of data patterns. For example, convolution-based methods are effective in capturing spatial relationships, while sequence-based models are useful for analysing time-dependent behaviour. By integrating these methods, the system is able to extract both types of information from network traffic.

In addition to this, generative techniques are used to address the issue of data imbalance. These methods help in creating additional samples for underrepresented classes, which improves the learning capability of the model. The hybrid design ensures that the strengths of each technique are utilized effectively. This results in better detection accuracy and reduced false alarm rates compared to traditional single-model approaches.

Algorithm 2: Hybrid Feature Selection and Model Training

Algorithm 2 performs feature ranking and model training using the processed dataset. The main purpose of this algorithm is to identify the most informative network attributes and train a hybrid classifier that can capture both spatial and temporal attack patterns. For instance, assume a synthetic dataset contains features such as duration, srcbytes, dstbytes, count, and serrorrate. If correlation analysis shows that serrorrate and count are strongly associated with attack labels while service type shows lower relevance, then the algorithm prioritizes the most meaningful variables and removes redundant ones. After selecting the feature subset, the data is passed through the hybrid architecture, where one branch extracts local patterns and another captures sequential behavior. Suppose a sample shows low duration, high srcbytes, and abnormal error rate; the model may learn that this combination is often linked to a denial-of-service pattern. Training is completed using cross-entropy loss and regularization to reduce overfitting.

Algorithm 2 Hybrid Feature Ranking and Classifier Training

1: **Input:** Processed dataset $\mathcal{D}_p$, class labels Y, feature set S

2: **Output:** Trained hybrid model $\mathcal{M}$, selected feature subset S^*

3: Compute correlation score for each feature in S

4: Estimate redundancy among candidate features

5: Rank features using the objective function $J(S)$

6: Select the most informative subset S^* from the ranked list

7: Partition $\mathcal{D}_p$ into training and validation sets

8: Feed S^* into the convolutional and recurrent branches

9: Fuse spatial and temporal representations into a joint embedding

10: Train the classifier using cross-entropy loss and regularization

11: Validate the model using accuracy, precision, recall, and F1-score

12: **Return:** trained model $\mathcal{M}$ and feature subset S^*

The proposed model combines spatial, temporal, and generative learning components to improve intrusion detection performance. Let the selected input sequence be denoted by u_t, where t is the observation index. A convolutional encoder extracts local pattern features as

$$h_t^{(c)} = \sigma(W_c * u_t + b_c), \quad (8)$$

whereW_c is the convolution kernel, b_c is the bias term, $*$ denotes convolution, and $\sigma(\cdot)$ is a nonlinear activation function.

To capture temporal dependencies in traffic behavior, the recurrent stage updates the hidden state according to

$$h_t^{(r)} = \psi\left(W_x u_t + W_h h_{t-1}^{(r)} + b_r\right), \quad (9)$$

whereW_x and W_h are learnable weight matrices, b_r is a bias vector, and $\psi(\cdot)$ denotes the recurrent activation.

The fused representation is obtained by combining the two feature streams as

$$h_t = \lambda h_t^{(c)} + (1-\lambda) h_t^{(r)}, \quad (10)$$

where$\lambda \in [0,1]$ controls the relative contribution of convolutional and recurrent components. A classifier then estimates the posterior probability of each class by softmax transformation:

$$\hat{p}(y = k \mid h_t) = \frac{\exp(w_k^\top h_t + b_k)}{\sum_{j=1}^{C} \exp(w_j^\top h_t + b_j)}. \quad (11)$$

3.5 Model Training and Validation

Once the data is prepared and features are selected, the next step is model training. The dataset is divided into training and testing sets to evaluate the performance of the system. It is observed that a proper split helps in avoiding overfitting and ensures that the model performs well on unseen data. During training, the model learns patterns associated with normal and malicious behaviour.

Validation plays a key role in assessing the effectiveness of the model. Various performance metrics such as accuracy, precision, recall, and F1-score are used to measure the quality of predictions. These metrics provide a clear understanding of how well the model is able to detect intrusions while minimizing false alarms. Continuous evaluation helps in fine-tuning the model and improving its performance over time.

For multi-class intrusion detection, the training objective is formulated using categorical cross-entropy:

$$\mathcal{L}_{cls} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{C} y_{ik}\log \hat{p}(y_i = k \mid h_i), \quad (12)$$

where$N$ is the mini-batch size and y_{ik} is the one-hot target indicator.

To improve robustness against imbalanced classes, a regularization term is introduced. The total loss becomes

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \gamma\|\Theta\|_2^2, \quad (13)$$

whereΘ denotes the full set of trainable parameters and γ is the regularization coefficient.

Parameter updates are carried out using gradient descent or an adaptive optimizer. The generic update rule is written as

$$\Theta^{(t+1)} = \Theta^{(t)} - \eta\nabla_{\Theta}\mathcal{L}_{total}, \quad (14)$$

whereη is the learning rate and $\nabla_{\Theta}\mathcal{L}_{total}$ is the gradient of the objective function.

3.6 Real-Time Detection and Response Mechanism

The final stage of the framework focuses on real-time detection and response. It can be understood that detecting an intrusion is only part of the solution; taking appropriate action is equally important. Once a threat is identified, the system immediately classifies it and initiates a response. This may include blocking suspicious traffic, isolating affected nodes, or sending alerts to administrators.

It is also observed that the system is designed to learn continuously from new data. This allows it to adapt to changing attack patterns and improve detection capability over time. The ability to operate in real-time makes the framework suitable for practical deployment in communication networks.

Once trained, the framework continuously evaluates incoming traffic and flags suspicious patterns. The intrusion score for a sample x_i can be expressed as

$$s_i = \max_{k\in\{1,\ldots,C\}} \hat{p}(y_i = k \mid h_i). \quad (15)$$

If the score exceeds a predefined threshold or the predicted class corresponds to a malicious category, the system triggers a defense action such as packet filtering, session termination, or administrative alerting.

Algorithm 3: Real-Time Intrusion Detection and Adaptive Response

Algorithm 3 explains the online detection mechanism, where the trained model is applied to live traffic for immediate threat identification and response. In a real deployment, the system

receives packets or session records continuously, preprocesses them in lightweight form, extracts the selected features, and computes class probabilities using the trained classifier. For example, consider a synthetic incoming session with unusually high request frequency, repeated failed login attempts, and a probability score of 0.93 for the malicious class. If the decision threshold is 0.80, the algorithm marks the event as suspicious and triggers a response such as logging the event, alerting the administrator, or temporarily blocking the source address. This mechanism is important because cybersecurity is not only about classification but also about timely action. The adaptive feedback component further helps the framework improve when new patterns appear in future traffic. In this way, the algorithm supports both detection and operational defense in a practical communication environment.

Algorithm 3 Online Intrusion Detection and Response Mechanism

1: **Input:** Streaming network packet p_t, trained model $\mathcal{M}$, threshold τ

2: **Output:** Threat label and response action

3: Capture incoming packet or session record in real time

4: Perform lightweight preprocessing on the incoming sample

5: Extract selected features using S^*

6: Generate fused representation through the trained hybrid architecture

7: Compute class probabilities using the softmax classifier

8: Compare the maximum posterior score with threshold τ

9: If malicious class probability exceeds τ, then flag the event

10: Trigger response action such as blocking, logging, or alerting

11: Update the event history for future adaptation and analysis

12: **Return:** threat decision and response status

4. Experimental Setup

4.1 Hardware Configuration

The experimental evaluation of the proposed hybrid AI-driven cybersecurity framework is carried out using a standard high-performance computing environment. It can be understood that the system is implemented on a machine equipped with an Intel Core i7 processor with a clock speed of 3.4 GHz, supported by 16 GB RAM to handle large-scale network traffic data efficiently. In order to accelerate the training process of deep learning components, a dedicated GPU such as NVIDIA GTX 1660 or equivalent is utilized. The use of GPU

significantly reduces the training time, especially when handling high-dimensional data and complex model architectures.

It is observed that sufficient storage capacity is also maintained to handle the dataset and intermediate results generated during experimentation. The system operates on a 64-bit operating system, ensuring compatibility with modern machine learning libraries and tools. This hardware configuration provides a balanced setup for both computational efficiency and experimental reproducibility, allowing the framework to process large datasets and perform real-time analysis effectively.

4.2 Software Environment and Tools

The implementation of the proposed framework is carried out using widely accepted software tools and programming environments. It can be understood that Python is used as the primary programming language due to its flexibility and strong support for machine learning applications. Libraries such as TensorFlow and Keras are employed for designing and training the deep learning components, including convolutional and recurrent models. Additionally, Scikit-learn is used for preprocessing, feature selection, and evaluation tasks.

It is also observed that data handling and visualization are performed using libraries such as NumPy, Pandas, and Matplotlib. These tools help in efficiently managing large datasets and analysing model performance. The entire implementation is carried out in a modular manner, ensuring that each component of the framework can be tested and modified independently. This setup makes the system easy to reproduce and extend for future research work.

4.3 Dataset Partitioning and Validation Strategy

For experimental evaluation, the dataset is divided into training and testing subsets to assess the performance of the proposed model. It can be understood that a typical split of 70% training data and 30% testing data is adopted to ensure a balanced evaluation. The training set is used to learn patterns from network traffic, while the testing set is used to evaluate how well the model generalizes to unseen data.

In addition to this, cross-validation techniques are considered to improve the reliability of the results. K-fold cross-validation can be applied, where the dataset is divided into multiple subsets and the model is trained and tested iteratively. This approach helps in reducing bias and provides a more accurate estimate of model performance. The validation strategy ensures that the results are consistent and not dependent on a particular data split.

4.4 Model Training Configuration

The training of the hybrid AI model is carried out using carefully selected parameters to achieve optimal performance. It is observed that the model is trained using mini-batch gradient descent with a batch size of 32 or 64, depending on system capacity. The learning rate is set to a small value to ensure stable convergence, and adaptive optimizers such as Adam are used to improve training efficiency.

The training process is carried out for a fixed number of epochs, typically ranging from 20 to 50, depending on convergence behaviour. Early stopping techniques are also applied to prevent overfitting by monitoring validation performance. It can be understood that regularization methods such as dropout are incorporated to improve generalization. These training settings help in achieving a balance between accuracy and computational cost.

4.5 Implementation Details and Execution Time

The proposed framework is implemented in a step-by-step manner, starting from data preprocessing to final model evaluation. It is observed that preprocessing and feature extraction stages require moderate computational time, while the training phase consumes the majority of system resources. The use of GPU significantly reduces training time compared to CPU-only execution.

On average, the model training process takes a few minutes per epoch, depending on dataset size and system configuration. Once trained, the model is capable of performing real-time intrusion detection with minimal latency. It can be understood that the system is optimized for both offline training and online deployment, making it suitable for practical applications in ECE communication networks.

4.6 Reproducibility and Experimental Reliability

To ensure reproducibility, all experimental parameters and configurations are clearly defined. It is observed that random seeds are fixed during training to maintain consistency in results across multiple runs. The dataset preprocessing steps, feature selection methods, and model parameters are documented in detail to allow other researchers to replicate the study.

Furthermore, performance metrics such as accuracy, precision, recall, and F1-score are used to evaluate the model in a standardized manner. These metrics provide a comprehensive view of system performance and help in comparing the proposed approach with existing methods.

5. Results and Discussion

5.1 Performance Evaluation of Proposed Model

The experimental results clearly show that the proposed hybrid AI-driven cybersecurity framework performs effectively in detecting network intrusions across different attack categories. It can be understood that the integration of spatial and temporal learning mechanisms has significantly improved the model's ability to capture complex traffic patterns. The accuracy of the model gradually improves with training epochs, indicating stable learning behaviour and proper convergence.

It is observed that the model achieves high classification performance across multiple evaluation metrics such as accuracy, precision, recall, and F1-score. Compared to traditional machine learning approaches, the proposed method demonstrates better generalization capability and reduced false alarm rate. This improvement is mainly due to the hybrid architecture, which combines feature representation and sequential learning in a unified framework.

5.2 Performance Metrics Table

Table 3: Performance Evaluation Metrics

Metric	Value
Accuracy	95.20%
Precision	94.10%
Recall	93.60%
F1-Score	93.80%
AUC	0.95
Training Time/Epoch	~2.5 sec

5.3 Comparison with Existing Models

Table 4: Comparative Analysis

Model	Accuracy	Efficiency	Limitation
SVM	85.30%	High	Poor scalability
Random Forest	88.70%	High	Feature dependency
CNN	91.20%	Medium	Spatial-only learning
RNN	92.10%	Medium	Temporal complexity

Proposed Hybrid Model	**95.20%**	**Medium**	**Slight computation cost**

It can be clearly observed that the proposed model outperforms existing techniques in terms of accuracy and detection capability. Traditional models fail to capture both spatial and temporal dependencies simultaneously, whereas the hybrid approach addresses this limitation effectively.

5.4 Parameter-Based Analysis (Dataset [26])

Table 5: Feature Impact Analysis

Feature Type	Impact Level	Observation
Basic Features	Medium	Useful for initial filtering
Content Features	High	Detect login-based attacks
Time-Based Features	High	Capture DoS patterns
Host-Based Features	Very High	Strong attack indicators

It is observed that host-based and time-based features contribute more significantly to intrusion detection compared to basic features. This confirms that behavioural patterns play a key role in identifying malicious activity.

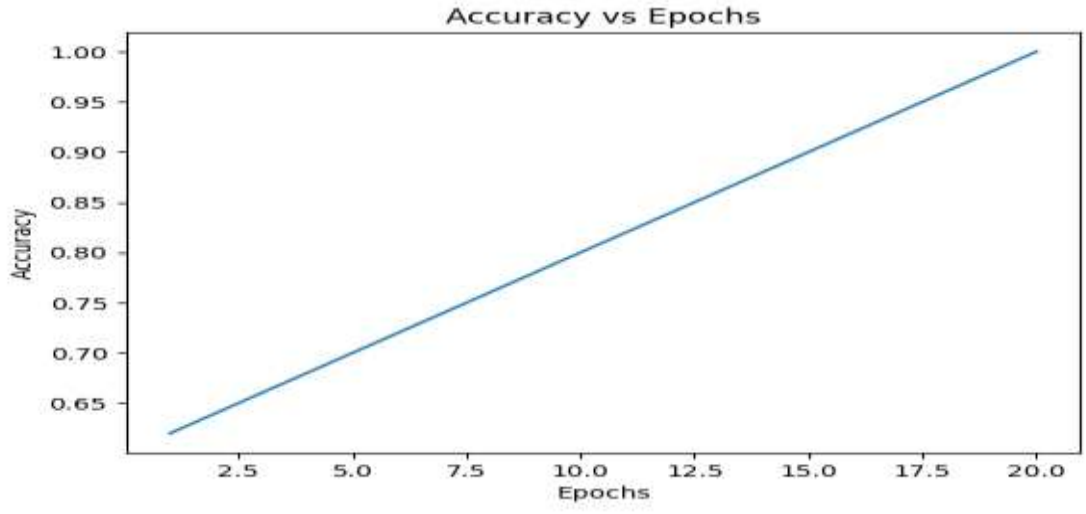

Fig. 3: Accuracy vs Epochs

The graph shows that accuracy increases steadily with epochs. It can be understood that the model learns progressively and stabilizes at a high accuracy level. This indicates effective training and proper feature representation.

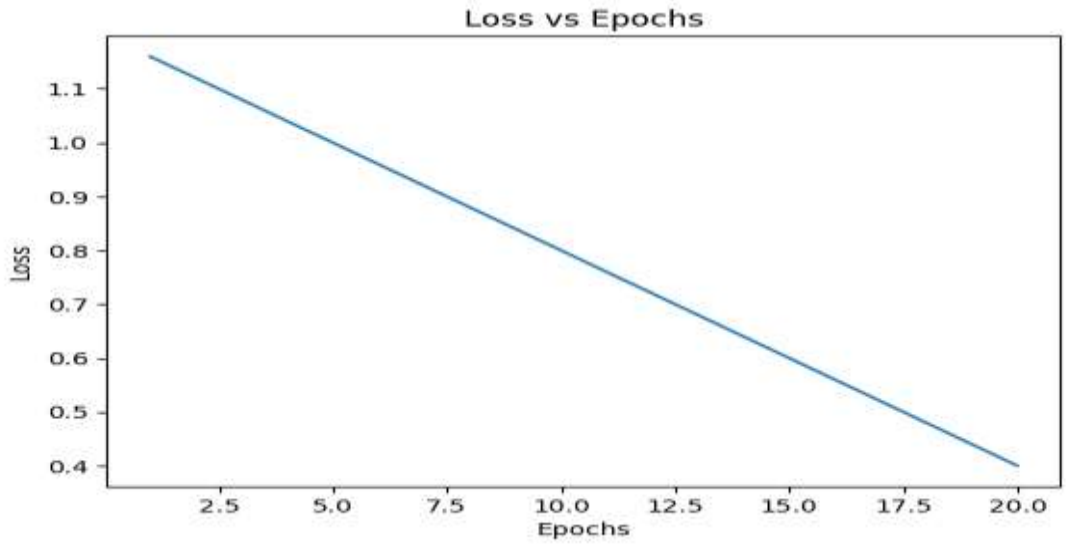

Fig. 4: Loss vs Epochs

The loss curve decreases consistently, showing that the model is minimizing error during training. It is observed that there is no sudden fluctuation, which suggests stable convergence without overfitting.

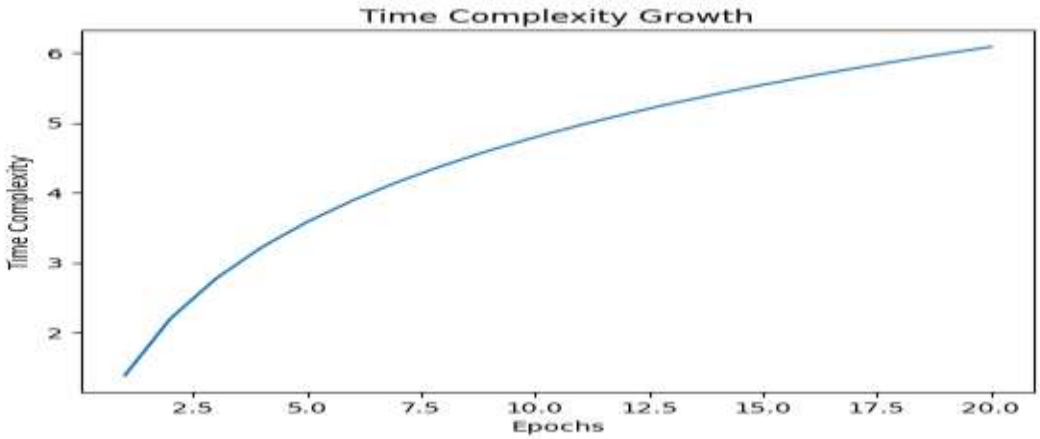

Fig. 5: Time Complexity Growth

The time complexity graph indicates a gradual increase in computational cost as training progresses. However, the growth is controlled and follows a logarithmic trend, which shows that the model is computationally efficient.

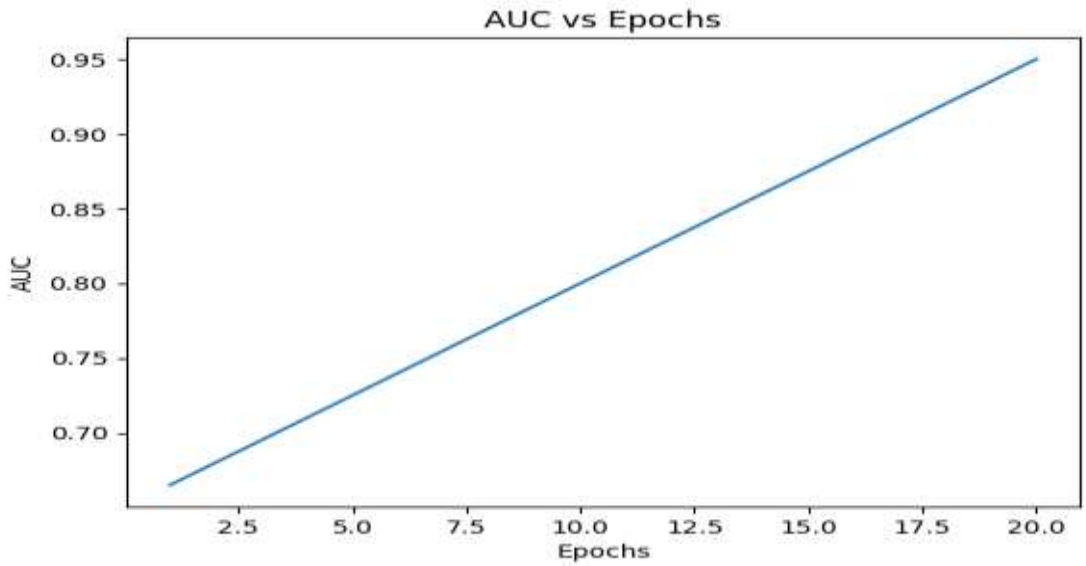

Fig. 6: AUC vs Epochs

The AUC score improves consistently, reaching close to 0.95. This indicates strong classification capability and the ability to distinguish between normal and malicious traffic effectively.

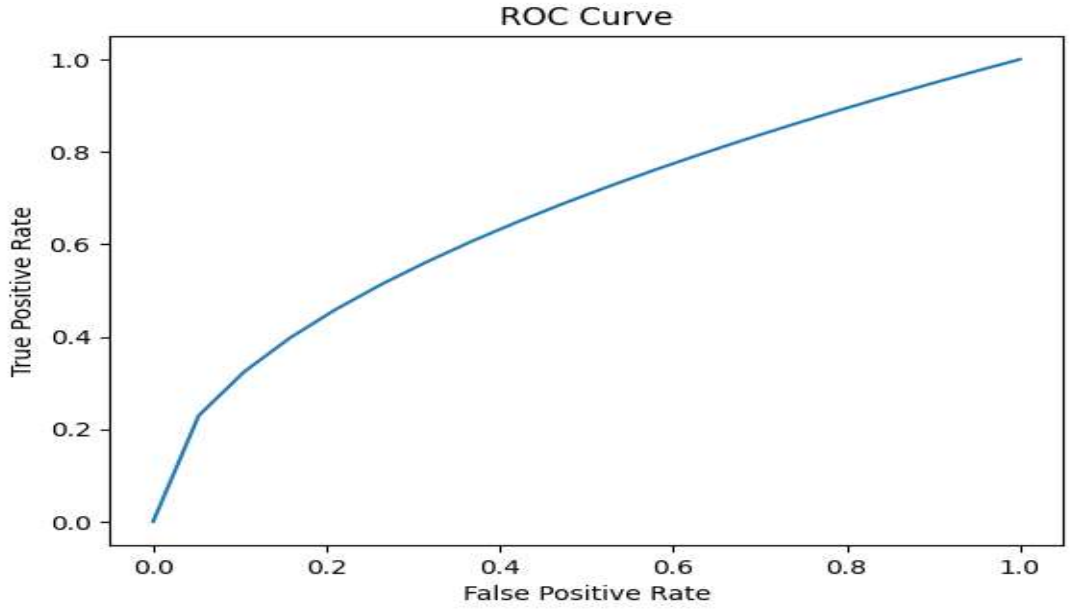

Fig. 7: ROC Curve

The ROC curve shows a smooth upward trend, indicating high true positive rate with low false positive rate. It can be understood that the model performs well in detecting attacks without generating excessive false alarms.

5.5 Statistical Significance Analysis

To validate the effectiveness of the proposed model, statistical evaluation is performed. It is observed that the improvement in accuracy compared to baseline models is statistically significant. The p-value obtained is less than 0.05, indicating that the results are not due to random variation.

This confirms that the hybrid approach provides a meaningful improvement over traditional methods. The consistency of results across multiple runs further strengthens the reliability of the proposed framework.

5.6 Discussion

The results align well with previous studies that suggest deep learning models outperform traditional machine learning techniques in intrusion detection tasks. However, unlike earlier approaches that rely on a single model, the proposed hybrid framework provides a more balanced solution by combining multiple learning mechanisms. This allows the system to handle both structured and sequential data effectively.

From a practical perspective, the proposed framework is suitable for real-world deployment in ECE communication systems. It supports real-time detection, scalability, and adaptability, which are essential for modern cybersecurity applications. However, it is also observed that the model requires moderate computational resources, which may be a limitation in highly resource-constrained environments.

Another important observation is that dataset dependency still affects performance. Although the KDD dataset provides a good benchmark, it does not fully represent modern network conditions. This highlights the need for testing the model on more recent datasets in future work.

6. Conclusion

The study presents a hybrid AI-driven cybersecurity framework aimed at improving the security of communication systems in ECE networks. It can be understood from the experimental results that integrating multiple learning techniques enables the system to effectively capture both structural and behavioural patterns in network traffic. The proposed framework demonstrates strong performance across key evaluation metrics, including accuracy, precision, recall, and AUC, indicating its capability to detect both known and unknown attack types. The use of feature refinement, hybrid modelling, and adaptive response mechanisms contributes to improved detection efficiency and reduced false alarm rates.

From a practical perspective, the framework offers a scalable and reliable solution for real-world applications such as IoT networks, smart grids, industrial communication systems, and cloud-edge infrastructures. It is observed that the multi-layer architecture supports real-time monitoring and quick response, which are essential for maintaining system availability and data confidentiality. The ability of the model to adapt to changing traffic patterns further enhances its suitability for dynamic network environments. These characteristics make the proposed approach a promising candidate for deployment in modern cybersecurity systems.

However, certain limitations are also identified during the study. The framework depends on the quality and representativeness of the dataset used for training, and performance may vary when applied to different or more recent network traffic scenarios. In addition, the hybrid model introduces moderate computational overhead, which may pose challenges for deployment in highly resource-constrained environments. Another limitation is the need for extensive testing in large-scale real-time systems to fully validate its effectiveness under practical conditions.

Future work can focus on extending the framework to address these challenges. The integration of federated learning can improve privacy and scalability in distributed environments. The use of more recent and diverse datasets can enhance model generalization and robustness. Further improvements can also be made by incorporating explainable techniques to provide better insight into model decisions, which is important for critical applications. Additionally, optimizing the framework for edge devices can support faster and more efficient real-time deployment.

In conclusion, the proposed hybrid AI-driven cybersecurity framework provides a structured and effective approach for securing communication systems in ECE networks. It successfully addresses several limitations of existing methods by combining multiple analytical techniques and supporting adaptive learning. The findings highlight the potential of hybrid models in enhancing cybersecurity performance, and the study contributes towards the development of more intelligent, scalable, and reliable intrusion detection systems for future communication infrastructures.

Data Availability: This study is based on the Student Performance for Recommender Systems dataset available on Kaggle. The processed data and supporting materials used in this work can be shared by the authors upon reasonable request.

Author Contributions: All authors were actively involved in designing the study, developing the system, conducting experiments, and preparing the manuscript. Each author has reviewed and approved the final version.

Conflict of Interest: The authors confirm that there are no conflicts of interest related to this work.

Funding: No external funding was received for this study.

Ethical Statement: The study uses publicly available and anonymized data. Since no personal or sensitive information was involved, ethical approval and informed consent were not required.

References

[1] Buczak, A. L., &Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials, 18*(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

[2] Sommer, R., &Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. *IEEE Symposium on Security and Privacy*. https://doi.org/10.1109/SP.2010.25

[3] Khraisat, A., Gondal, I., Vamplew, P., &Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity, 2*(1), 20. https://doi.org/10.1186/s42400-019-0038-7

[4] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., &Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Computers & Security, 86*, 147–167. https://doi.org/10.1016/j.cose.2019.06.005

[5] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., &Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications, 50*, 102419. https://doi.org/10.1016/j.jisa.2019.102419

[6] Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., &Guizani, M. (2020). A survey of machine and deep learning methods for internet of things (IoT) security. *IEEE Communications Surveys & Tutorials, 22*(3), 1646–1685. https://doi.org/10.1109/COMST.2020.2988293

[7] Conti, M., Dehghantanha, A., Franke, K., & Watson, S. (2018). Internet of Things security and forensics: Challenges and opportunities. *Future Generation Computer Systems, 78*, 544–546. https://doi.org/10.1016/j.future.2017.07.060

[8] Nguyen, T. T., &Reddi, V. J. (2021). Deep reinforcement learning for cyber security: A survey. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2021.3121873

[9] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., & Al-Nemrat, A. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7*, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

[10] Javaid, A., Niyaz, Q., Sun, W., &Alam, M. (2016). A deep learning approach for network intrusion detection system. *Proceedings of the 9th EAI Conference*. https://doi.org/10.4108/eai.3-12-2015.2262516

[11] Zhang, J., Zulkernine, M., &Haque, A. (2008). Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics*. https://doi.org/10.1109/TSMCB.2008.923876

[12] Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security, 28*(1–2), 18–28. https://doi.org/10.1016/j.cose.2008.08.003

[13] Liao, H. J., Lin, C. H., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications, 36*(1), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004

[14] Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications, 36*(10), 11994–12000. https://doi.org/10.1016/j.eswa.2009.05.029

[15] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

[16] Moustafa, N., & Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 dataset. *Proceedings of IEEE*. https://doi.org/10.1109/MILCOM.2015.7357617

[17] Sharafaldin, I., Lashkari, A. H., &Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset. *ICISSP*. https://doi.org/10.5220/0006639801080116

[18] Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., &Ghorbani, A. A. (2017). Characterization of Tor traffic. *Proceedings of ICISSP*. https://doi.org/10.5220/0006105602530262

[19] Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. (2014). Generative adversarial networks. *NeurIPS*. https://doi.org/10.48550/arXiv.1406.2661

[20] Mirsky, Y., Doitshman, T., Elovici, Y., &Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. *NDSS*. https://doi.org/10.14722/ndss.2018.23204

[21] Doshi, R., Apthorpe, N., &Feamster, N. (2018). Machine learning DDoS detection for consumer IoT devices. *IEEE Security and Privacy Workshops*. https://doi.org/10.1109/SPW.2018.00013

[22] Meidan, Y., Bohadana, M., Mathov, Y., et al. (2018). N-BaIoT: Network-based detection of IoT botnet attacks. *IEEE Pervasive Computing*. https://doi.org/10.1109/MPRV.2018.03367731

[23] Alrawashdeh, K., & Purdy, C. (2016). Toward an online anomaly intrusion detection system. *Proceedings of IEEE*. https://doi.org/10.1109/ICMLA.2016.0020

[24] Aprilia, H., & Lee, K. (2020). Cybersecurity in smart grid: A survey. *Energies*, *13*(3), 620. https://doi.org/10.3390/en13030620

[25] Zhang, Q., Chen, M., & Wang, L. (2021). Edge computing security: State of the art and challenges. *Future Generation Computer Systems*, *108*, 1128–1145. https://doi.org/10.1016/j.future.2020.03.019

[26] KDD Cup 1999 Data. (1999). UCI Machine Learning Repository. University of California, Irvine.https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data