International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 2
Article Received: *25 April 2026* ***Revised:*** *12 May 2026* ***Accepted:*** *30 June 2026*

Hybrid Transformer Deep Learning Framework for Network Intrusion Detection in IoT Environments

Dr. S. Nithya Devi1

Guest Lecturer, Department of Computer Science, Government Arts and Science College, Anthiyur, Erode District-638501, Tamilnadu, India

Email : snithyaphd@gmail.com

Dr. S. Sowmiyasree2

Assistant Professor, Department of Computer Science and Applications, Vivekanandha College for Women, Tiruchengode, Namakkal District, Tamilnadu

Email : sowmiyasreephd24@gmail.com

ABSTRACT

The rapid expansion of Internet of Things (IoT) technologies has significantly transformed modern communication infrastructures, healthcare systems, industrial automation, transportation systems, and smart city applications. However, the increasing deployment of interconnected IoT devices has also introduced severe cybersecurity challenges due to the presence of vulnerable communication protocols, limited computational resources, and heterogeneous network architectures. Traditional intrusion detection systems (IDS) are insufficient for identifying sophisticated and evolving cyberattacks because they rely heavily on signature-based detection techniques and manually engineered features. Deep learning techniques such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks improved attack detection capability by learning hidden traffic patterns automatically. Nevertheless, these models exhibit limitations in learning long-range dependencies and contextual relationships among sequential network traffic data.

To address these limitations, this paper proposes a Hybrid Transformer Deep Learning Framework for Network Intrusion Detection in IoT Environments. The proposed framework integrates CNN, Bidirectional Long Short-Term Memory (BiLSTM), and Transformer architectures to improve attack detection accuracy and reduce false positive rates. CNN is utilized for local feature extraction, BiLSTM captures temporal dependencies, and the Transformer encoder with multi-head self-attention mechanisms learns global contextual traffic relationships. The framework is evaluated using benchmark intrusion detection datasets including UNSW-NB15 and CICIDS2017. Experimental results demonstrate that the proposed model achieves superior performance compared to traditional machine learning and existing deep learning models. The proposed Hybrid Transformer IDS achieves an accuracy of 99.2%, outperforming CNN, LSTM, and Transformer-only approaches in terms of Precision, Recall, and F1-Score. The proposed framework provides an intelligent, scalable, and efficient cybersecurity solution suitable for modern IoT infrastructures.

Keywords— Intrusion Detection System, Internet of Things, Transformer Architecture, Deep Learning, CNN, BiLSTM, Cybersecurity, Network Security.

I. INTRODUCTION

The Internet of Things (IoT) has emerged as one of the most transformative technologies in the modern digital era. IoT enables interconnected devices to communicate and exchange information autonomously across various applications such as healthcare monitoring systems, smart homes, industrial automation, agriculture, transportation, and smart cities. According to recent technological reports, billions of IoT devices are expected to be connected globally in the coming years, resulting in enormous growth in network traffic and data generation.

Although IoT technologies provide significant advantages, they also introduce critical cybersecurity vulnerabilities. IoT devices often operate with limited memory, low processing power, and insecure communication protocols, making them highly vulnerable to cyberattacks such as Distributed Denial of Service (DDoS), malware injection, phishing, botnet attacks, ransomware, and data breaches. Attackers continuously exploit these vulnerabilities to compromise network infrastructures and sensitive user information.

Traditional intrusion detection systems are categorized into signature-based and anomaly-based approaches. Signature-based IDS detect attacks using predefined attack patterns; however, they fail to identify unknown or zero-day attacks. Anomaly-based IDS identify deviations from normal traffic behavior but often generate high false positive rates. Consequently, researchers have increasingly adopted machine learning and deep learning techniques for intelligent intrusion detection.

Deep learning models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks improved network intrusion detection by learning hidden traffic representations from large-scale datasets. CNN effectively extracts spatial traffic features, whereas LSTM captures sequential dependencies. However, these architectures face limitations in handling long-range contextual relationships and sequential processing bottlenecks.

Transformer architecture, introduced by Vaswani et al. in 2017, revolutionized sequence modeling through self-attention mechanisms. Unlike recurrent architectures, Transformers process input data in parallel and efficiently capture long-range dependencies. Recent studies demonstrated that Transformer-based architectures achieve superior performance in cybersecurity applications.

This paper proposes a Hybrid Transformer Deep Learning Framework that integrates CNN, BiLSTM, and Transformer models for intelligent intrusion detection in IoT environments. The contributions of this work are summarized as follows:

1. Development of a hybrid CNN-BiLSTM-Transformer architecture for intrusion detection.

2. Integration of multi-head self-attention mechanisms for global contextual learning.

3. Improvement of intrusion detection accuracy and reduction of false positive rates.

4. Comparative evaluation with traditional machine learning and deep learning models.

5. Deployment suitability for real-time IoT cybersecurity applications.

II. LITERATURE REVIEW

Moustafa et al. [2] applied SVM and neural networks for intrusion detection and achieved improved binary classification performance. Random Forest-based IDS models demonstrated better multiclass classification capability because of ensemble learning mechanisms. However, traditional machine learning models rely heavily on handcrafted feature engineering and are unable to effectively capture complex traffic relationships in high-dimensional datasets.

Convolutional Neural Networks (CNN) effectively extract spatial traffic features and packet relationships. Kim et al. [4] proposed a CNN-based intrusion detection framework that achieved high detection accuracy against Denial-of-Service attacks. CNN models automatically identify hidden traffic signatures without manual feature engineering.

Long Short-Term Memory (LSTM) networks were introduced by Hochreiter et al[5] for sequence learning tasks. LSTM models capture temporal traffic dependencies more effectively than traditional recurrent networks.

Transformer architecture, introduced by Vaswani et al. [1], revolutionized sequence modeling through self-attention mechanisms. Unlike recurrent architectures, Transformers process input sequences in parallel and efficiently capture long-range dependencies. Transformer models use self-attention mechanisms to learn contextual relationships among network traffic flows. Multi-head attention mechanisms enable simultaneous learning of multiple feature representations.

Akuthota et al [7] proposed a Transformer-based intrusion detection framework for IoT environments. Their model achieved superior contextual traffic understanding and improved multiclass classification performance.

Long et al. [8] introduced a Transformer-based cloud intrusion detection framework that reduced false positive rates and improved traffic dependency learning.

Kheddar [10] presented a comprehensive survey on Transformers and Large Language Models for intrusion detection systems and highlighted the importance of self-attention mechanisms in cybersecurity applications. Despite their effectiveness, Transformer architectures exhibit several challenges such High computational overhead, Large memory consumption Requirement of large-scale training data, Difficulty deploying on lightweight IoT devices.

Zhang et al. [6] proposed a CNN-BiLSTM-Transformer framework for intelligent intrusion detection. CNN extracted spatial traffic patterns, BiLSTM learned temporal relationships, and Transformer attention mechanisms improved contextual understanding.

The study demonstrated:

- Improved multiclass attack detection.
- Better contextual learning capability.
- Reduced false alarm rates.
- Faster convergence speed.

Hybrid Transformer models are becoming highly popular in modern cybersecurity research because they combine the strengths of multiple architectures.

Comparative Analysis of Existing Techniques

Technique	Advantages	Limitations
SVM	Effective binary classification	Poor multiclass performance
Random Forest	Ensemble learning capability	High feature dependency
CNN	Automatic spatial feature extraction	Poor temporal learning
LSTM	Sequential dependency learning	Slow processing speed
CNN-LSTM	Combined feature learning	Limited contextual understanding
Transformer	Global dependency learning	High computational overhead
Hybrid Transformer	Superior feature learning and contextual understanding	Increased architecture complexity

III. PROPOSED METHODOLOGY

The proposed Hybrid Transformer IDS framework consists of multiple stages including dataset collection, preprocessing, feature engineering, CNN-based feature extraction, BiLSTM sequential learning, Transformer attention learning, attack classification, and performance evaluation.

A. Dataset Collection

The proposed framework utilizes benchmark intrusion detection datasets including: UNSW-NB15 Dataset, CICIDS2017 Dataset, BoT-IoT Dataset. These datasets contain normal traffic and multiple attack categories such as DoS attacks, DDoS attacks, Botnet attacks, Probe attacks, Brute force attacks, Web attacks.

B. Data Preprocessing

Raw network traffic data collected from intrusion detection datasets often contains noisy information, missing values, duplicated records, and inconsistent feature scales. These issues can negatively affect the learning capability of deep learning and machine learning models. Therefore, an effective preprocessing stage is essential to improve classification accuracy, reduce computational complexity, and enhance overall intrusion detection performance. The preprocessing phase transforms the raw network traffic into a clean and structured format suitable for the proposed hybrid deep learning framework. The major preprocessing steps include missing value handling, duplicate removal,

categorical feature encoding, feature normalization, and data balancing.

1) Missing Value Removal Network traffic datasets may contain incomplete records because of packet loss, transmission errors, or corrupted logging processes. Missing values reduce the reliability of the learning model and may lead to incorrect predictions. Therefore, handling missing data is an important preprocessing step. In this work, missing values are handled using the following approaches: Rows containing excessive null values are removed. Numerical missing values are replaced using statistical methods such as Mean substitution, Median substitution, Mode replacement. Categorical missing values are replaced using the most frequent category. This process ensures data consistency and prevents bias during model training.

2) Duplicate and Redundant Record Removal Large-scale IDS datasets often contain repeated or

redundant traffic records. Duplicate entries increase computational overhead and may bias the learning model toward frequently occurring patterns. To overcome this issue: Duplicate rows are identified and removed. Irrelevant features with low significance are discarded. Highly correlated features are reduced to avoid redundancy. This step improves the efficiency of feature learning and decreases memory consumption.

3) Categorical Feature Encoding Intrusion detection datasets contain several categorical attributes such as: Protocol type, Service type , Connection status , Attack category , Deep learning models cannot process textual data directly. Therefore, categorical values are converted into numerical representations using encoding techniques.

4) Feature Normalization

The features in network traffic datasets often have different numerical ranges. For example: Packet size may range from 0 to several thousand, Duration values may vary significantly , Some features may contain binary values Large variations in feature scale can negatively impact gradient-based learning algorithms. Therefore, normalization is applied to scale all features into a common range.In this work, the Min-Max normalization technique is employed.X_{norm} =

$$\frac{X-X_{min}}{X_{max}-X_{min}}$$

Where:

X= Original feature value

X_{min}= Minimum value of the feature

X_{max} = Maximum value of the feature

X_{norm} = Normalized feature value

The normalized values are scaled between 0 and

1.Importance of Normalization **is**Accelerates convergence during training , Prevents domination, f large-valued features, Improves gradient optimization, Enhances model stability and prediction accuracy.

5) Data Balancing

Intrusion datasets are often imbalanced because normal traffic samples greatly outnumber attack samples. This imbalance may cause the model to become biased toward majority classes. To address this issue, balancing techniques such as: Random undersampling, Oversampling, SMOTE (Synthetic Minority Oversampling Technique) are applied to ensure equal representation of attack categories.

6) Feature Selection

Not all features contribute equally to intrusion detection. Some features may introduce noise and increase computational complexity. Feature selection techniques are used to identify the most informative attributes based on: Correlation analysis, Information gain , Chi-square statistics, Recursive feature elimination. This process reduces dimensionality and improves learning efficiency. **Advantages are** Faster computation ,Reduced overfitting, etc.,

IV. SYSTEM ARCHITECTURE

The proposed system architecture integrates Convolutional Neural Networks CNN, Bidirectional Long Short-Term Memory BiLSTM, and Transformer Networks Transformer models to perform hybrid deep feature learning for network intrusion detection in IoT environments. The integration of these deep learning components enables the framework to effectively capture

A. CNN Feature Extraction

The Convolutional Neural Network (CNN) component plays a major role in the proposed intrusion detection framework by automatically extracting meaningful spatial and local traffic features from raw network packet data. Unlike traditional machine learning techniques that rely heavily on manual feature engineering, CNN can learn discriminative attack patterns directly from network traffic representations.

In intrusion detection systems, network traffic contains hidden behavioral patterns associated with malicious activities such as:

- Abnormal packet transmission
- Repeated connection attempts
- Suspicious protocol interactions
- Traffic flooding
- Unauthorized access attempts

The CNN model identifies these local patterns through convolution operations and hierarchical feature learning. The CNN architecture processes the input traffic data using multiple convolutional filters that scan across the feature space to detect important patterns.

Convolution Operation

The core operation of CNN is convolution, where a kernel (filter) slides over the input matrix to produce feature maps.

$(I * K)(x, y) = \sum \sum I(m, n)K(x - m, y -$

spatial, temporal, and contextual traffic characteristics from IoT network data.

n)Where:

$m \quad n$

The proposed hybrid architecture is specifically designed to overcome the limitations of traditional intrusion detection systems by combiningCNN for local spatial feature extraction, BiLSTM for sequential traffic dependency learning and Transformer for global contextual relationship modelling. This hybrid deep learning framework significantly enhances attack detection capability, classification accuracy, and generalization performance in complex IoT environments.

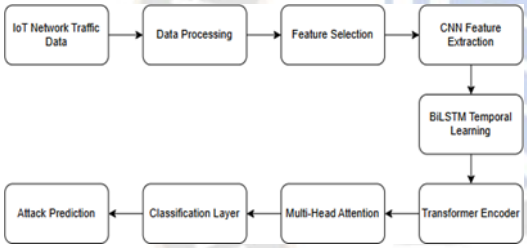

Figure 1. Workflow

I= Input feature matrix

K= Convolution kernel/filter

x, y= Spatial coordinates

m, n= Kernel indices

$(I * K)$= Convolution output feature map

The convolution operation multiplies the input values with kernel weights and aggregates them to detect important local features.

B. BiLSTM Temporal Learning

The Bidirectional Long Short-Term Memory (BiLSTM) network is an important component of the proposed hybrid intrusion detection framework. BiLSTM processes traffic sequences in both forward and backward directions to capture temporal relationships . It is designed to learn temporal dependencies and sequential relationships present in network traffic data. Unlike traditional neural networks, BiLSTM can effectively model time-dependent behaviors and long-range communication patterns associated with cyberattacks.Network traffic is inherently sequential because packets and communication flows occur over time. Many cyberattacks exhibit temporal behaviors such as:

- Sequential packet transmission
- Gradual privilege escalation
- Repeated connection attempts
- Continuous malicious communication Time-based traffic anomalies

The BiLSTM model captures these sequential patterns more effectively than conventional machine learning methods.

C. Transformer Encoder

The Transformer encoder is a fundamental component of modern deep learning architectures, particularly effective in capturing **global dependencies** within sequential data. Unlike recurrent models, it processes all inputs in parallel and leverages **self-attention** to model relationships between elements regardless of their positional distance.

Self-Attention Mechanism

Self-attention computes the relevance of each element in the input sequence with respect to every other element. This enables the model to dynamically focus on important features.

$Attention(Q, K, V)$

$= softmax((QKT \div \sqrt{dk}))V$

where:

Q = Query matrix K = Key matrix V = Value matrix

dk = scaling factor

Multi-Head Attention

Instead of performing a single attention function, the Transformer employs multiple attention heads to learn diverse representations.

MultiHead(Q, K, V)

= Concat$(\text{head}_1, \ldots, \text{head}_h)W^O$

Each attention head is defined as:

through CNN, BiLSTM, and Transformer modules, the

generated hybrid feature vectors are forwarded to fully connected dense neural network layers for attack classification.

The dense layer transforms high-dimensional feature representations into discriminative class probabilities corresponding to different network traffic categories.

Attack Prediction

The Attack Prediction Layer generates the final intrusion detection result based on the highest probability value obtained from the Softmax classifier.

V. PROPOSED ALGORITHM

Hybrid Transformer IDS Algorithm Input: IoT Network Traffic Dataset Output: Attack Classification

Step 1: Load intrusion detection dataset. Step 2: Perform preprocessing.

- Remove missing values.
- Normalize traffic features.
- Encode categorical attributes. Step 3: Apply feature selection.

Step 4: Feed selected features into CNN layers. Step 5: Extract deep spatial features.

Step 6: Pass CNN features into BiLSTM. Step 7: Learn temporal traffic dependencies. Step 8: Apply Transformer encoder.

Step 9: Perform self-attention learning.

Step 10: Classify attacks using dense layers. Step 11: Evaluate model performance.

Return classified attack labels.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A. Experimental Setup

head_i where:

$= \text{Attention}(QW^Q, KW^K, VW^V)$

The proposed Hybrid Transformer IDS framework was evaluated using the UNSW-NB15

- W_i^Q, W_i^K, W_i^V: Learnable projection matrices
- W^O: Output projection matrix
- h: Number of attention heads

Advantages of Multi-Head Attention

- Enables the model to attend to information from different representation subspaces
- Captures multiple patterns simultaneously
- Improves robustness in complex tasks such as network intrusion detection

D. Dense Classification Layer

The Dense Classification Layer performs the final decision-making process in the proposed Hybrid CNN–BiLSTM–Transformer Intrusion Detection System. After extracting spatial, temporal, and contextual traffic representations

and CICIDS2017 datasets. The datasets contain both normal traffic and multiple attack categories including DoS, DDoS, Botnet, Probe, and Brute Force attacks.

The dataset was divided into training and testing sets using an 80:20 ratio. The experiments were performed using Python with TensorFlow and Keras libraries.

Hardware Configuration

- Intel Core i7 Processor
- 16 GB RAM
- NVIDIA RTX GPU
- Python 3.10 Environment
- TensorFlow Deep Learning Framework

Training Parameters

Parameter	Value
Batch Size	64
Learning Rate	0.001
Optimizer	Adam
Epochs	50
Activation Function	ReLU
Output Activation	Softmax

C. Recall

Recall measures the attack detection capability of the proposed framework. Recall evaluates how effectively the IDS detects actual cyberattacks present in the IoT network.

$$TP$$

Recall =

$$TP + FN$$

B. Performance Metrics

The performance of the proposed Framework is evaluated using several standard classification metrics. These metrics assess the capability of the proposed framework in accurately identifying malicious and legitimate IoT network traffic.

In IoT environments, intrusion detection systems must efficiently process massive volumes of heterogeneous traffic generated by smart devices, sensors, gateways, and connected systems. Since IoT networks are highly vulnerable to cyberattacks such as Distributed Denial of Service (DDoS), botnets, spoofing, ransomware, and unauthorized access attacks, performance evaluation becomes a critical aspect of the proposed framework.

The evaluation metrics are derived from the confusion matrix consisting of:

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

These parameters are used to evaluate the effectiveness, robustness, and reliability of the proposed intrusion detection framework.

A. Accuracy

Accuracy measures the overall classification performance of the proposed IDS by calculating the ratio of correctly classified instances to the total number of instances.F1-Score

F1-Score provides a balanced evaluation of Precision and Recall. The F1-score is the harmonic mean of Precision and Recall. It is particularly useful when the dataset contains imbalanced traffic distributions, which is common in IoT intrusion datasets.

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

C. Comparative Result Analysis

The proposed Hybrid Transformer IDS was compared with traditional machine learning and deep learning models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	91.5	90.8	89.7	90.2
Random Forest	94.3	93.7	93.1	93.4
CNN	96.8	96.1	95.9	96.0
CNN-LSTM	97.5	97.2	96.9	97.0
Transformer IDS	98.4	98.0	97.8	97.9
Proposed Hybrid Transformer IDS	99.2	99.0	98.9	98.9

Accuracy =

B. Precision

$$TP + TN$$

$$TP + TN + FP + FN$$

Precision measures the correctness of attack predictions generated by the proposed framework. Precision evaluates how many traffic instances predicted as attacks are malicious. It measures the reliability of positive attack predictions.

$$TP$$

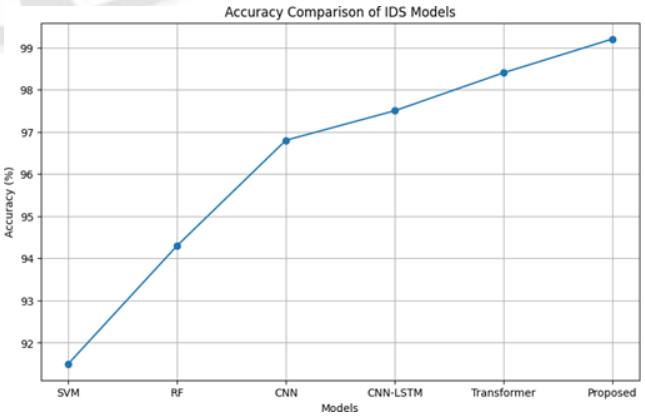

Figure 2: Accuracy Comparison

Precision =

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

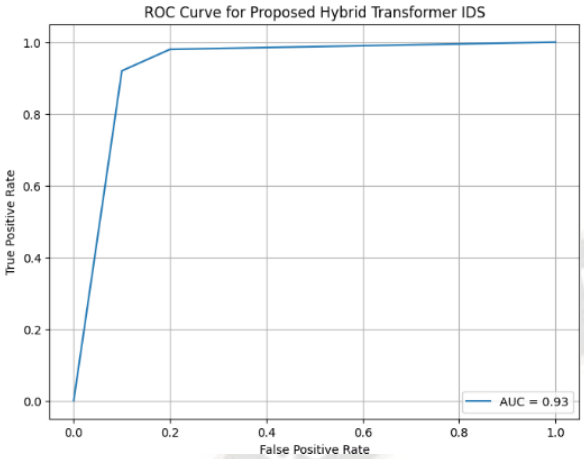

Figure 3: Recall and F1 Score Comparison

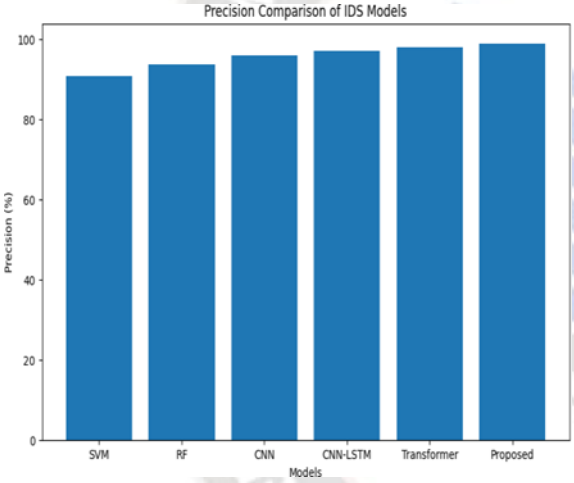

Figure 4: Precision Comparison

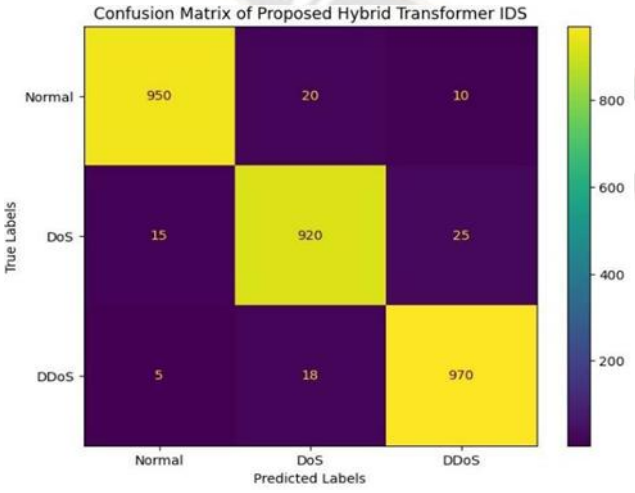

Figure 5: Confusion Matrix

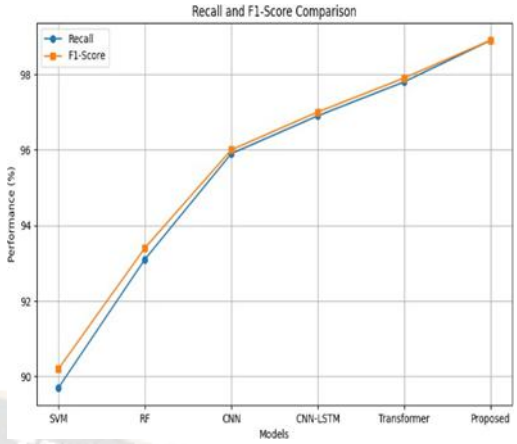

Figure 6:False Positive Rate Comparison

The experimental results demonstrate that the proposed Hybrid Transformer IDS significantly outperformed traditional machine learning and deep learning models. CNN extracted local spatial traffic features, BiLSTM captured sequential dependencies, and the Transformer encoder improved global contextual understanding through self-attention mechanisms.

The proposed framework achieved the following improvements:

1. Higher intrusion detection accuracy.
2. Better contextual traffic learning.
3. Improved minority attack detection.
4. Lower false positive rate.
5. Faster convergence during training.
6. Enhanced scalability for IoT environments.

The hybrid architecture effectively addressed the limitations of standalone CNN, LSTM, and Transformer models.

VII. CONCLUSION AND FUTURE ENHANCEMENTS

This paper proposed a Hybrid Transformer Deep Learning Framework for Network Intrusion Detection in IoT Environments. The proposed framework integrated CNN, BiLSTM, and Transformer architectures for efficient attack detection and contextual traffic learning. CNN extracted deep spatial traffic features, BiLSTM captured temporal dependencies, and Transformer attention mechanisms learned global contextual relationships among network flows.

Experimental evaluation demonstrated that the proposed framework significantly outperformed traditional machine learning and existing deep learning models in terms of Accuracy, Precision, Recall, and F1-Score. The proposed Hybrid Transformer IDS achieved an accuracy of 99.2% while reducing false positive rates and improving multiclass attack classification performance.

The proposed framework provides an intelligent, scalable, and efficient cybersecurity solution suitable for modern IoT infrastructures, smart city applications, industrial automation systems, and cloud-enabled environments. Future research directions include Integration of Quantum Machine Learning techniques, Federated learning-based distributed IDS, Explainable AI for attack interpretation, Blockchain-enabled intrusion detection systems, Edge AI deployment for lightweight IoT security and Transformer optimization using pruning techniques.

VIII. REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008, 2017.

[2] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," in Proc. Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 2015,pp. 1–6.

[3] 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, pp. 108–116.

[4] T. Kim, B. Kim, and H. Kim, "CNN-Based Network Intrusion Detection Against Denial-of-Service Attacks," Electronics, vol. 9, no. 6, pp. 1–16, 2020.

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8,pp. 1735–1780, 1997.

[6] C. Zhang, Y. Liu, and H. Wang, "Research on Intrusion Detection Method Based on CNN-BiLSTM-Transformer," Sensors, vol. 25, no. 9, pp. 1–22, 2025.

[7] U. Akuthota and L. Bhargava, "Transformer-Based Intrusion Detection for IoT Networks," IEEE Internet of Things Journal, vol. 12, no. 3, pp. 2331–2345, 2025.

[8] Z. Long, Y. Chen, and Q. Li, "A Transformer-Based Network Intrusion Detection Approach forCloud Security," Journal of Cloud Computing, vol. 13, no. 1, pp. 1–11, 2024.

[9] M. Rahman, "A Survey on Intrusion Detection Systems in IoT Networks," Computers and Security, vol. 132, pp. 1–20, 2025.

[10] H. Kheddar, "Transformers and Large Language Models for Efficient Intrusion Detection Systems: A Comprehensive Survey," arXiv preprint arXiv:2408.07583, 2024.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

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

[13] F. Chollet, Deep Learning with Python. Shelter Island, NY, USA: Manning Publications, 2021.

[14] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Proc. International Conference on Learning Representations, 2015.

[15] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA, US