Received: May 9, 2026. Revised: June 14, 2026.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Graph Attention-driven Transformer Model for Intelligent IoT Anomaly Classification

Anjitha K[1]* **A. Saritha[1]**

[1] *Department of Computer Science and Engineering*
Vels Institute of Science, Technology and Advanced Studies, Chennai, India
* Corresponding author's Email: anjithakmcchet@gmail.com

Abstract: The rapid expansion of Internet of Things (IoT) networks has increased the vulnerability of connected devices to sophisticated cyber threats, necessitating accurate and reliable anomaly classification mechanisms. Conventional machine learning and signature-based intrusion detection approaches often struggle to capture complex feature interactions and long-range dependencies present in heterogeneous IoT traffic. To address these challenges, this study proposes a hybrid deep learning framework that integrates a learnable Graph Neural Network (GNN) with a Transformer encoder for intelligent IoT anomaly classification. The GNN branch dynamically learns inter-feature relationships through a trainable soft adjacency matrix, enabling adaptive structural representation learning without predefined graph connectivity. Simultaneously, the Transformer branch employs multi-head self-attention to model contextual and long-range dependencies among traffic features. The embeddings generated by both branches are fused to produce a discriminative representation for multi-class anomaly classification. The proposed framework was evaluated on the CIC-IoT 2023 dataset containing seven major attack categories, including DDoS, DoS, spoofing, brute force, reconnaissance, web attacks, and Mirai attacks. Experimental results achieved 97.51% accuracy, 97.56% precision, 97.25% recall, and 97.40% F1-score, with a misclassification rate of 2.49%. Ablation studies verified the effectiveness of the learnable adjacency mechanism and the hybrid architecture, while sensitivity analysis identified eight attention heads as the optimal configuration. Robustness analysis across multiple random seeds demonstrated stable performance, and external validation on the IoT-23 dataset achieved 96.24% accuracy, confirming strong generalization capability. The results indicate that the proposed GNN–Transformer framework provides a robust, scalable, and effective solution for anomaly classification in modern IoT security environments.

Keywords: Internet of things, Deep learning, Anomaly detection, Graph neural network, Transformer cyberattacks.

1. Introduction

The IoT has rapidly become an integral component of modern digital ecosystems, interconnecting billions of devices that generate, transmit, and process data in real time [1]. These networks extend across domains such as manufacturing, smart city management and health care, where automation and intelligent decision-making are crucial. Fig. 1 shows the IoT-based smart environments. By allowing continuous monitoring and responsive control, IoT systems contribute significantly to efficiency, safety, and convenience. However, their size, diversity, and dependence on resource-constrained devices expose IoT systems to operational faults and malicious exploitation [2]. Within this context, anomaly detection, the ability to detect irregularities in behavior or data patterns that differ from the expected norm, plays an important role towards enhancing IoT reliability and security [3].

The significance of anomaly detection lies in its preventive and protective capacity. Even small irregularities, if unnoticed, turn into major system failures. For example, a misbehaving sensor in a healthcare monitoring system led to an inaccurate diagnosis, while anomalies in industrial IoT disrupt production processes, cause equipment damage or endanger worker safety. In energy networks, unusual

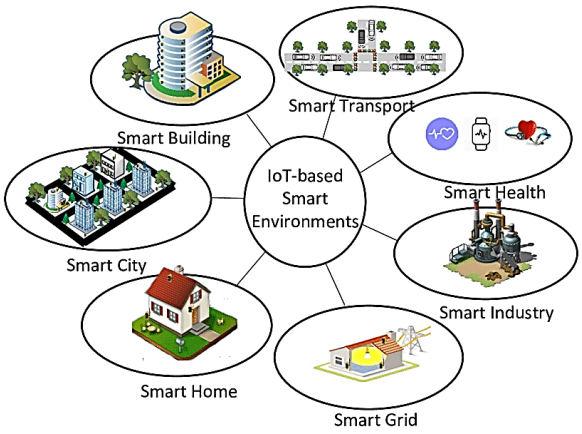

Figure. 1 IoT-based smart environments

consumption patterns or abnormal grid behaviour indicate attacks that compromise stability and affect millions of users.

Early and accurate anomaly detection is essential for preventing cascading failures and maintaining the reliability of IoT ecosystems [4,5]. Owing to their large scale, dynamic topology, and heterogeneous device connectivity, IoT environments are vulnerable to diverse cyber threats, including DDoS, DoS, spoofing, reconnaissance, web-based, brute-force, and malware-driven attacks such as Mirai [6–9]. These attacks can disrupt services, compromise sensitive information, and exploit resource-constrained devices, posing significant risks to critical applications. Consequently, there is an increasing need for intelligent and adaptive anomaly detection mechanisms capable of identifying both known and emerging attack patterns. Traditional anomaly detection approaches, while simple and interpretable, often struggle to model the complex and evolving characteristics of IoT traffic, resulting in reduced detection accuracy and increased false alarms.

Recent advances in artificial intelligence, particularly DL, have shown remarkable promise in overcoming these limitations. Deep models excel at capturing nonlinear relationships and automatically extracting features from raw data streams, offering superior accuracy and scalability compared to traditional techniques [10-12]. However, most existing AI-based solutions treat IoT traffic either as flat sequences or as independent features, overlooking the dual nature of IoT data where both structural relationships and temporal dependencies are critical. For instance, IoT devices interact through networked connections that exhibit graph-like structures, while traffic flows evolve sequentially, reflecting contextual dependencies over time. A framework that jointly model these structural and temporal aspects is therefore essential for effective anomaly detection. This work proposes a hybrid DL framework that combines GNNs and Transformers to jointly capture structural and sequential characteristics of IoT traffic data. The major objectives of the suggested model are:

- To develop a hybrid GNN-Transformer method for effective anomaly detection in IoT security.
- To enable the dynamic learning of feature relationships through a soft adjacency matrix.
- To combine structural and sequential patterns for higher accuracy, scalability, and robustness.

The rest of this paper is structured as follows: A review of anomaly detection is given in Section 2. The methodology is shown in Section 3, and the results of the suggested model are explained in Section 4. Section 5 provides concluding observations and future research.

2. Related works

Rodriguez et al. [13] proposed a two-stage IoT anomaly classification framework to improve safety and dependability in industrial processes. The framework combined an unsupervised anomaly detection model with a machine learning-based classification module. A Conv1D autoencoder was employed for anomaly detection using sliding-window validation, while a Transformer-based classifier was used to distinguish failures from cyberattacks. Evaluated on a conveyor-belt testbed dataset containing failures and attack scenarios, the model achieved an F1-score of 0.91, precision of 0.95, and recall of 0.88. Although effective in differentiating failures and attacks, the framework exhibited limitations in correctly classifying previously unseen anomalies. Abudurexiti et al. [14] developed an unsupervised anomaly detection framework for Industrial IoT systems by integrating convolutional feature extraction, an improved Temporal Convolutional Network, and a Kolmogorov-Arnold Networks-based Variational Autoencoder. The model captured both local and long-term temporal dependencies, while a dynamic scoring mechanism and top-k thresholding were used for anomaly identification. Experimental results on multiple benchmark datasets demonstrated superior robustness and detection performance compared with existing unsupervised methods. Furthermore, Explainable Artificial Intelligence (XAI) techniques were incorporated to improve interpretability. However, the computational complexity of the framework may limit its suitability for large-scale real-time deployment.

Katib et al. [15] introduced a TinyML-enabled deep learning framework for real-time anomaly detection in IoT environments. The approach combined Z-score normalization, fennec fox optimization-based feature selection, gradient least mean squares with BiLSTM classification, and Jaya-based hyperparameter optimization. The model achieved high detection accuracy and demonstrated effectiveness for predictive maintenance applications. Nevertheless, its performance depended heavily on dataset characteristics, feature quality, and computationally intensive optimization procedures. Wahab et al. [16] developed CNN-, DNN-, and Transformer-based models for 12-class intrusion detection using the CIC-IoT-2023 dataset. The proposed framework achieved a maximum accuracy of 93.0%, demonstrating effective multiclass (12 class) attack detection. However, the study was limited to DDoS-centric attack categories and did not investigate cross-dataset robustness or real-world deployment feasibility.

Gummadi et al. [17] proposed an Explainable Artificial Intelligence (XAI)-based anomaly detection framework that combined AI-driven anomaly detection with feature importance analysis. Multiple single and ensemble learning models were evaluated using IoT manufacturing sensor data and the N-BaIoT botnet dataset. To improve interpretability, seven XAI techniques, including SHAP and LIME, were employed to identify influential features. The framework effectively detected anomalies and explained model decisions; however, its evaluation was limited to specific datasets, restricting broader generalizability.

Tahir et al. [18] developed a machine learning-based framework for IoT anomaly detection and adaptive defense. Several classifiers, including Decision Tree, Random Forest, Gradient Boosting, and Support Vector Machine, were evaluated on IoT network traffic data. Gradient Boosting achieved the best performance with an accuracy of 89.34%, while SVM exhibited poor generalization and overfitting. Despite promising results, scalability and overfitting remained significant challenges. Altulaihan et al. [19] introduced an intrusion detection framework using Decision Tree, Random Forest, KNN, and SVM classifiers with Genetic Algorithm and Correlation-Based Feature Selection. Experiments on the IoT-ID20 dataset showed that Decision Tree and Random Forest achieved perfect classification performance when combined with Genetic Algorithm-based feature selection. However, the high computational cost of feature optimization, possible overfitting, and limited validation reduced practical applicability. Rajaan et al. [20] proposed a hybrid anomaly detection model for smart home IoT systems by combining Random Forest and Autoencoder models through a decision-fusion strategy. The framework achieved 95.6% accuracy and reduced false positives by 20% compared with individual models. Nevertheless, maintaining dual detection models increased computational complexity and deployment overhead.

Zulfiqar et al. [21] proposed DeepDetect, a hybrid deep learning framework integrating CNN, Gated Recurrent Unit (GRU), and BiLSTM networks for IoT anomaly detection. CNN layers extracted spatial traffic features, GRUs captured feature correlations while mitigating vanishing gradients, and BiLSTM layers modeled bidirectional temporal dependencies. Evaluated on the NSL-KDD dataset, the model achieved high accuracy in both binary and multi-class classification tasks. However, the reliance on the imbalanced NSL-KDD dataset raised concerns regarding evaluation fairness and generalizability. Khan et al. [22] developed a machine learning-based anomaly detection framework for healthcare IoT networks using the CIC-IoT dataset. The study evaluated several supervised algorithms, including Deep Neural Networks, Random Forest, Perceptron, AdaBoost, and Logistic Regression. Feature reduction, correlation analysis, and SMOTE-based balancing were employed during preprocessing. Random Forest consistently achieved the best performance while reducing computational complexity. Nevertheless, the framework struggled to distinguish certain attack categories, such as spoofing and reconnaissance attacks, and depended on synthetic data balancing techniques.

Albasir et al. [23] proposed an off-device anomaly detection framework that transformed power consumption signals into time-frequency representations using Constant-Q Transformation and Histogram of Oriented Gradients features. A CNN classifier was subsequently employed for anomaly classification. Experimental evaluation on multiple malware and DDoS datasets achieved approximately 88% accuracy and 85% F-score. Despite its effectiveness, the framework depended heavily on fixed sampling rates and was sensitive to hardware heterogeneity, limiting deployment flexibility. Mutambik et al. [24] introduced IoT-FIDS, a lightweight flow-based intrusion detection system that utilized communication behavior and flow-level packet characteristics to identify anomalous traffic. Evaluated on the BoT-IoT and UNSW-NB15 datasets, IoT-FIDS demonstrated high detection accuracy while maintaining low computational overhead. However, its reliance on predefined traffic profiles reduced adaptability to dynamic IoT

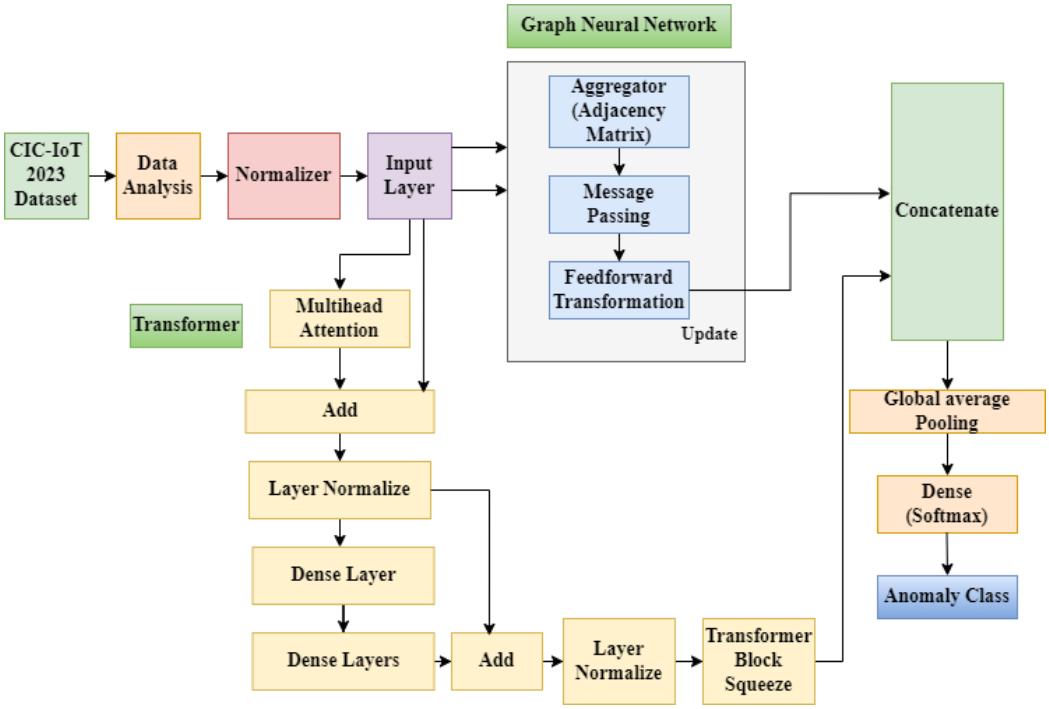

Figure. 2 Hybrid GNN-Transformer classification model

```
df.head(10)
```

ate	Srate	Drate	fin_flag_number	syn_flag_number	rst_flag_number	...	Std	Tot size	IAT	Number	Magnitue	Radius	Covariance	Variance	Weight	label
838	47.659838	0	0	0	0	...	0.000000	42.00	83124528.63	9.5	9.165151	0.000000	0.000000	0.00	141.55	DoS
296	35.836296	0	0	0	0	...	0.000000	54.00	83076264.65	9.5	10.392305	0.000000	0.000000	0.00	141.55	DoS
886	56.537886	0	0	0	0	...	190.084256	208.77	83124627.55	9.5	15.845730	268.910781	129729.387000	0.29	141.55	DoS
108	0.775108	0	0	0	0	...	0.000000	54.00	83075907.40	9.5	10.392305	0.000000	0.000000	0.00	141.55	DoS
837	0.665837	0	0	0	0	...	0.000000	42.00	83128762.77	9.5	9.165151	0.000000	0.000000	0.00	141.55	DoS
240	105.691240	0	0	0	0	...	0.000000	42.00	83149345.50	9.5	9.165151	0.000000	0.000000	0.00	141.55	DoS
537	16.342537	0	0	0	0	...	0.000000	42.00	83124764.57	9.5	9.165151	0.000000	0.000000	0.00	141.55	DoS
397	2129.606397	0	0	0	0	...	0.136929	50.42	83106988.55	9.5	10.004226	0.190714	0.616537	0.03	141.55	DoS
747	24.947747	0	0	0	0	...	0.000000	42.00	83132290.84	9.5	9.165151	0.000000	0.000000	0.00	141.55	DoS
799	2.999799	0	0	0	0	...	0.000000	54.00	83034127.20	9.5	10.392305	0.000000	0.000000	0.00	141.55	DoS

Figure. 3 Sample records from CIC IoT 2023 dataset

environments and encrypted communications.

Muñoz et al. [25] developed an unsupervised machine learning framework for real-time anomaly detection in heterogeneous IoT infrastructures. The system effectively identified and filtered anomalous packets within a large-scale university IoT deployment. While the approach improved data reliability, the unsupervised nature of the framework may limit its ability to differentiate rare legitimate behaviors from true anomalies. Shao et al. [26] proposed a Fast Johnson-Lindenstrauss Transform Bloom Filter (FJLT-BF) framework for anomaly detection in streaming IoT environments. By combining hashing techniques, Bloom filters, and Winner-Take-All competitive learning, the framework achieved approximately 95% detection accuracy on KDD and Credit datasets. However, its dependence on threshold tuning and pre-training constrained adaptability to evolving traffic patterns. Rahim et al. [27] introduced several hybrid models combining Logistic Regression, Gradient Boosting variants, and CNNs for anomaly detection and face recognition in smart home IoT environments. Among the evaluated models, LR-HGBC-CNN achieved the best anomaly detection accuracy of 94%. Despite improved performance, the framework exhibited strong dependency on dataset-specific characteristics, limiting its scalability and generalization to unseen environments.

2.1 Research gap

Although anomaly detection in IoT systems has seen substantial progress, several critical gaps remain. Many models are evaluated on limited or single datasets, which restricts their ability to generalize across heterogeneous IoT environments and diverse device ecosystems [13]. A key challenge lies in handling unseen anomalies, as several frameworks detect irregularities but often misclassify them as normal, reducing reliability in real-world deployments [14]. Deep learning and hybrid approaches, while effective, often involve higher computational requirements, which makes them less practical for real-time or large-scale applications in resource-constrained devices [15]. In addition, the use of imbalanced or synthetically balanced datasets may bias performance evaluation and hinder robustness against evolving attack strategies [22]. Some methods remain highly sensitive to device heterogeneity, where variations in hardware or power consumption behaviors undermine consistency [23]. Furthermore, approaches that depend on static flow profiles or limited protocol coverage struggle in dynamic IoT environments with encrypted communications [24]. These persistent limitations underscore the need for adaptive, lightweight, and generalizable anomaly detection frameworks that can operate across diverse and evolving IoT domains [18, 21, 25].

3. Materials and methods

The proposed hybrid model for anomaly classification in IoT security follows a DL framework integrating GNNs and Transformers, leveraging both the structural and sequential representations of IoT traffic data, as illustrated in Fig. 2. The raw CIC-IoT 2023 dataset with seven major threat categories is pre-processed with several operations, including label encoding of the categorical labels and standardization of the numerical features. This ensures uniform feature scaling, which also leads to consistency in the data in each learning stage. The normalized input features are then passed through two parallel paths.

```
df.describe()
```

	flow_duration	Header_Length	Protocol Type	Duration	Rate	Srate	Drate	fin_flag_number	syn_flag_number	rst_flag_number	...	AVG
count	350084.000000	3.500840e+05	350084.000000	350084.000000	3.500840e+05	3.500840e+05	350084.0	350084.000000	350084.000000	350084.000000	...	350084.000000
mean	124.339460	5.551630e+05	11.673425	89.874981	1.616805e+04	1.616805e+04	0.0	0.017973	0.063125	0.030667	...	418.871171
std	1208.083106	1.256745e+06	11.164552	39.449744	1.492109e+05	1.492109e+05	0.0	0.132853	0.243188	0.172414	...	567.041866
min	0.000000	0.000000e+00	0.000000	0.000000	0.000000e+00	0.000000e+00	0.0	0.000000	0.000000	0.000000	...	42.000000
25%	0.042346	1.080000e+02	6.000000	64.000000	8.585118e+00	8.585118e+00	0.0	0.000000	0.000000	0.000000	...	63.622869
50%	5.614385	1.893245e+04	7.100000	69.900000	3.929662e+01	3.929662e+01	0.0	0.000000	0.000000	0.000000	...	130.134051
75%	52.690727	4.334516e+05	12.400259	105.672251	1.735904e+02	1.735904e+02	0.0	0.000000	0.000000	0.000000	...	578.000000
max	99685.656390	9.831391e+06	47.000000	255.000000	3.145728e+06	3.145728e+06	0.0	1.000000	1.000000	1.000000	...	11650.000000

Figure. 4 Statistical summary of CIC IoT 2023 dataset

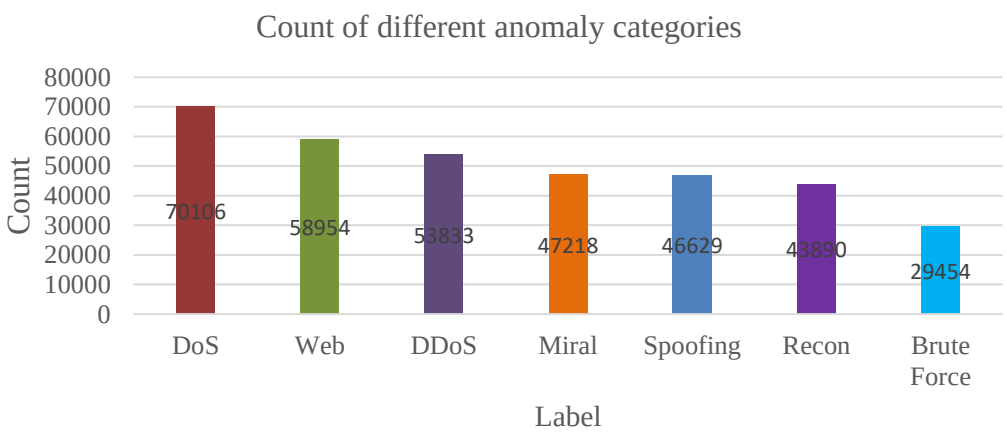

Figure. 5 Count of different anomaly categories

The first branch consists of a GNN, unlike traditional GNNs that use a predefined adjacency matrix as a static input, the proposed model dynamically learns a soft adjacency matrix during training, allowing flexible inter-feature connectivity and enabling each feature to adaptively capture self-dependence as well as cross-feature interactions.

The adjacency driven message passing is then followed by feedforward operations to update node embeddings and capture both global and local structural dependencies. The second branch uses a Transformer block to model temporal patterns and contextual relationships in the traffic flows. The input passes through a multi-head self-attention mechanism (MHSA), allowing the model to learn and capture long-range dependencies. Next, residual connections, followed by layer normalization and dense layers are sequentially applied within the transformer block to enhance learning stability and representational power. The transformer block outputs contextual embeddings that complement the GNN-derived structural features. The two different branches of embeddings are concatenated together to produce a joint representation. This combined representation is applied to a global average pooling layer, followed by a dense softmax classifier that predicts the probability of each one of the anomaly classes. By incorporating the GNN's advantages in capturing the graph-structured relationships, with the Transformer's ability in modelling sequential dependencies, the proposed hybrid architecture achieves robust and accurate anomaly classification across a variety of attacks in IoT environments.

3.1 Dataset description

The suggested model utilized the CIC IoT 2023 dataset, which is a real -time benchmark dataset for large-scale cyberattacks in IoT environments [28]. This dataset is widely recognized for its application in developing and testing security analytics within realistic IoT operations. It incorporates 33 distinct attack scenarios carried out within an IoT network topology consisting of 105 heterogeneous devices. These attacks are grouped into 7 categories such as Recon, DoS, Brute Force, Web, DDoS, Mirai and Spoofing. Each attack is generated by compromised IoT devices acting as adversaries against other devices within the network. Sample dataset is depicted in Fig. 3 and its statistical summary is illustrated in Fig. 4.

The class-wise distribution of anomalies in the CIC-IoT dataset is presented in Fig. 5. The dataset covers a wide range of cyberattacks, with the DoS class containing 70,106 instances, followed by Web (58954), DDoS (53833), and Mirai botnet (47218) traffic. Additionally, substantial records are available for spoofing (46629), reconnaissance (43890), and brute force (29454) categories. This comprehensive coverage across multiple attack types ensures that the dataset provides a diverse basis for training and evaluating anomaly detection models.

3.2 Exploratory data analysis (EDA)

An EDA was conducted to investigate the data properties and distribution of the features of the dataset before model development. The feature distribution plots illustrated in Fig. 6 show that many features of the continuous variables like flow duration, header length, flow rate, and packet statistics are generally right-skewed patterns. It generally signifies that most IoT flows were short and lightweight, with fewer cases of longer-duration flows or heavier data packet traffic which largely constituted outliers or anomalous events. Other features such as the minimum, maximum, mean, and variance exhibited sharp peaks at lower values but also contained wider tails, indicating occasional bursts of heavy traffic.

To explore interdependencies, a correlation heatmap was generated. The heatmap shows strong positive correlations among packet-and-byte-based features, indicating inherent redundancy in flow-level traffic metrics, while duration-related features show associations with selected flag counts, reflecting their influence on session dynamics. Protocol-level attributes such as HTTP, HTTPS, TCP, and UDP indicated measurable relationships with flow statistics, whereas categorical indicators including Telnet, SMTP, and IRC showed minimal correlation, providing complementary information. These observations support the multidimensionality of the dataset and highlight the importance of advanced learning models that are able to demonstrate both correlated and independent patterns for reliable anomaly detection. As demonstrated in the EDA, the CIC-IoT dataset contains a mixture of skewed numerical distributions, distinct binary protocol activations, and a variety of correlations across the features. These patterns demonstrate the importance of utilizing advanced representation learning techniques that recognize both statistical variability and structural dependencies to ensure accurate anomalous classification in IoT environments.

3.3 Data preprocessing

The CIC-IoT 2023 dataset, consisting of seven major anomaly classes, undergoes multiple

preprocessing steps to ensure suitability for deep learning models. Initially, the raw CIC-IoT 2023 records were examined for duplicate entries and invalid records. Duplicate traffic flows were removed during the data cleaning stage to prevent redundancy and potential bias during model training. After data cleaning and duplicate checking, 46 traffic-flow features were retained as input variables, namely: flow_duration, Header_Length, Protocol Type, Duration, Rate, Srate, Drate, fin_flag_number,syn_flag_number,rst_flag_number, psh_flag_number,ack_flag_number,ece_flag_number,cwr_flag_number,ack_count,syn_count, fin_count, urg_count, rst_count, HTTP, HTTPS, DNS, Telnet, SMTP, SSH, IRC, TCP, UDP, DHCP, ARP, ICMP, IPv, LLC, Tot sum, Min, Max, AVG, Std, Tot size, IAT, Number, Magnitue, Radius, Covariance, Variance, and Weight. The attack label was used as the target variable. The preprocessing steps include label encoding of categorical labels and feature scaling of numerical attributes. For multi-class classification, each attack class is mapped into a unique integer value ranging from 0 to 6 using label encoding. The attack categories were encoded using Scikit-learn LabelEncoder as follows: 0 = BruteForce, 1 = DDoS, 2 = DoS, 3 = Mirai, 4 = Recon, 5 = Spoofing, and 6 = Web. This operation is mathemattically expressed as in Eq. (1).

$$y_{encoded} = LabelEncoder(y) \quad (1)$$

where y denotes the original categorical attack label and $y_{encoded}$ represents the encoded integer label. To maintain uniformity across features and enhance model convergence, all numerical attributes were standardized using the StandardScaler technique. This transformation rescales each feature value $x$into a standardized value x', as expressed in Eq. (2).

$$x' = \frac{x-\mu}{\sigma} \quad (2)$$

where $\mu$and σdenote the mean and standard deviation of the corresponding feature, respectively. To prevent information leakage during model evaluation, the StandardScaler was fitted exclusively on the training subset. The mean and standard deviation obtained from the training data were subsequently used to transform the validation and testing subsets. No information from the testing data was used during scaler fitting. Standardization ensures that features with different scales contribute proportionally during model training, thereby improving optimization stability and learning performance. The original class distribution of the CIC-IoT 2023 dataset was preserved throughout the experiments. After preprocessing, the dataset was partitioned into training and testing subsets using a stratified 80:20 split implemented through the train_test_split() function with random_state = 42. Stratification was employed to preserve the original distribution of anomaly classes across both subsets, thereby minimizing sampling bias and ensuring reliable model evaluation. The resulting split produced 280,067 training samples and 70,017 testing samples. The train-test split was performed prior to model training and evaluation. During model training, 10% of the training subset was reserved for validation using the validation_split parameter, resulting in 252,060 samples for model training and 28,007 samples for validation. The validation subset was used exclusively for monitoring training performance and hyperparameter selection, while the independent testing subset was reserved for final evaluation. Also, the testing subset remained completely isolated throughout preprocessing, feature scaling, hyperparameter selection, and model optimization procedures to ensure an unbiased assessment of model performance. The class-wise distribution of samples in the training and testing sets is presented in Table 1.

Table 1. Class-wise distribution of training and testing samples after stratified 80:20 split

Attack Category	Total Samples	Training Samples	Testing Samples
DoS	70,106	56,085	14,021
Web	58,954	47,163	11,791
DDoS	53,833	43,066	10,767
Mirai	47,218	37,775	9,443
Spoofing	46,629	37,303	9,326
Recon	43,890	35,112	8,778
Brute Force	29,454	23,563	5,891
Total	**350,084**	**280,067**	**70,017**

3.4 Model development

The proposed hybrid model is designed to exploit both structural and sequential dependencies in IoT traffic data for anomaly detection. The GNN branch models feature interactions as graph structures, dynamically learning a soft adjacency matrix and using message passing to capture relational patterns. In parallel, the transformer leverages multi-head self-attention to extract temporal and contextual features from traffic flows. The embeddings generated by both branches are concatenated and passed through a dense softmax classifier, enabling accurate multi-class classification of IoT attack categories.

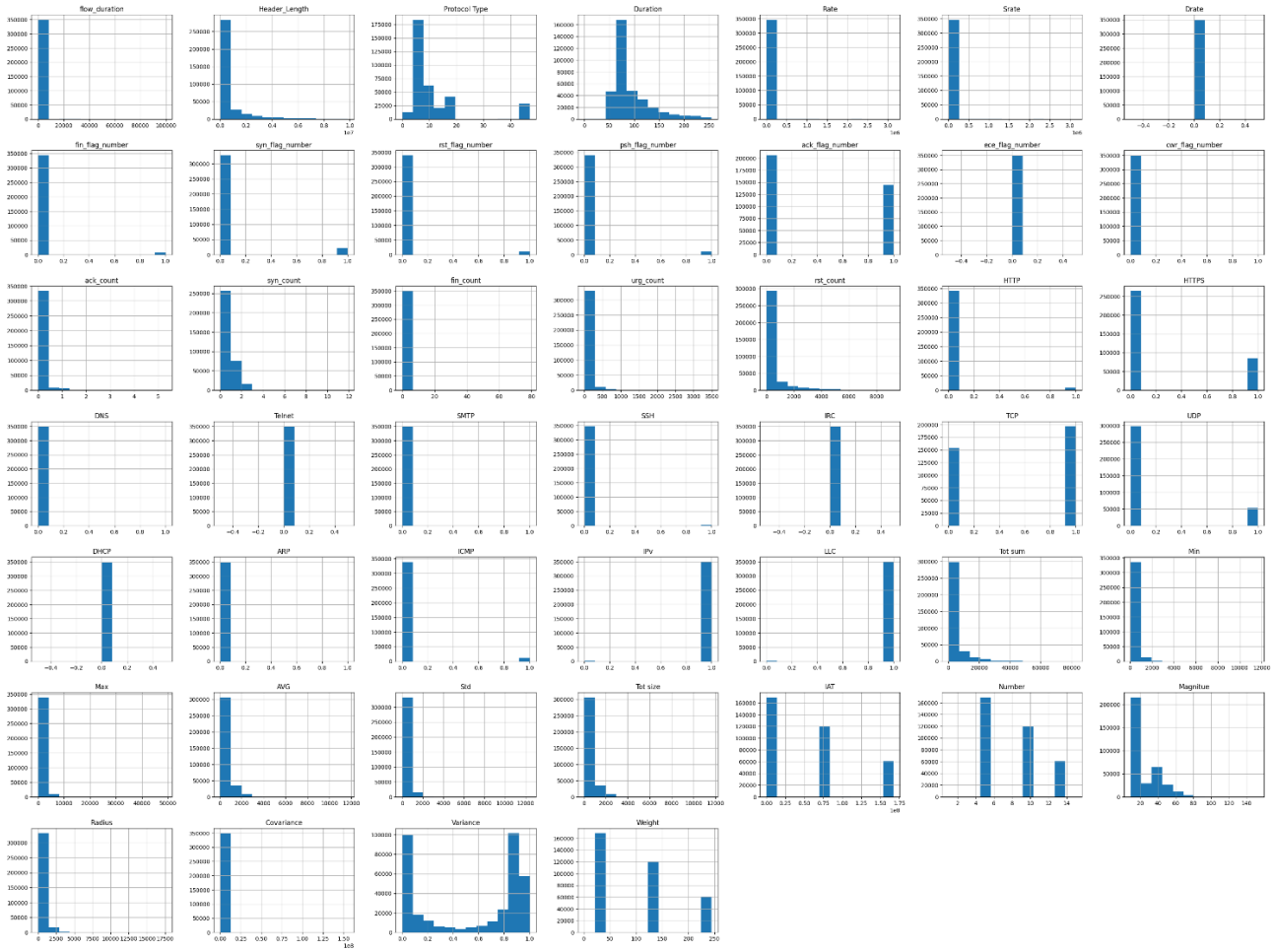

Figure. 6 Feature distribution

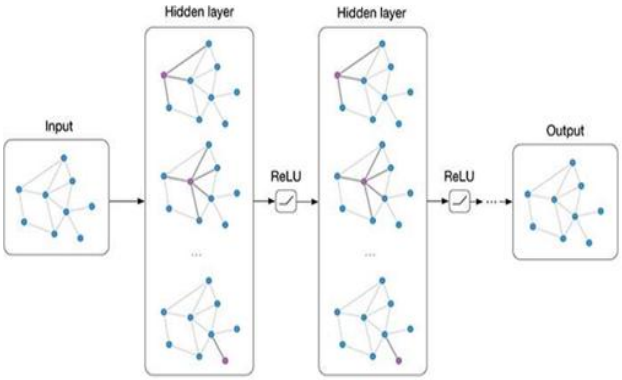

Figure. 7 Graph Neural Network architecture

3.4.1. GNN

GNNs are a class of DL models specifically designed to operate on data represented as graphs, such as data structures consisting of nodes or vertices and edges (links between nodes). In contrast to traditional NNs that work on fixed-size inputs like sequences or images, GNNs naturally handle non-Euclidean data where relationships among entities are irregular, unordered, and variable in size. In a graph, nodes represent entities and edges represent relationships [29, 30]. GNNs leverage this connectivity to make predictions not only based on an individual node's features but also by aggregating and learning from its neighbors and their neighbors in a recursive fashion, as shown in Fig. 7. In GNN, learning occurs by updating the representation, such as the embedding of each node, based on its own features and the features of its neighboring nodes. This process is performed iteratively at each layer of the network, and is known as message passing. The two major steps in the training of a GNN are: aggregation and update.

The first step of the message passing mechanism is called aggregation. During each step, each node collects information from its immediate neighbors to form a summary representation of its local neighborhood. A node is denoted by v and its set of neighboring nodes as $\mathcal{N}(v)$. At layer l, each neighbor $u \epsilon \mathcal{N}(v)$ has an embedding $h_u^{(l)}$, which represents the learned features of that node at the current layer. To aggregate information from these neighbors, the GNN applies an aggregation function to all the embeddings $h_u^{(l)}$ for nodes $u \epsilon \mathcal{N}(v)$. This operation is mathematically expressed as in Eq. (3).

$$m_v^{(l)} = Aggregrate\left(\left\{h_u^{(l)} | u \in \mathcal{N}(v)\right\}\right) \quad (3)$$

where $m_v^{(l)}$ is the aggregated message that node v receives from its neighbors. Once a node v has received and aggregated message messages from its neighbors, the next step in a GNN is to update the node's own representation. This is mathematically expressed as in Eq. (4).

$$h_v^{(l+1)} = Update\ (h_v^{(l)}, m_v^{(l)}) \quad (4)$$

where $h_v^{(l+1)}$ is the updated embedding at the next layer, $m_v^{(l)}$ is the aggregated message from the neighbors of nodes v and $h_v^{(l+1)}$ denotes the updated embedding at the next year. The purpose of this step is to combine the node's own information with the information received from its neighborhood to form a new, richer representation.

3.4.2. Transformer

The transformer block is designed to capture global dependencies and contextual interactions among features in input signals, as illustrated in Fig. 8. It employs a self-attention mechanism, which enables the model to assign varying levels of importance to various positions within the sequence without using recurrent structures [31]. Given an input feature vector $x > \epsilon \mathbb{R}^F$, the block first expands the tensor into a sequence of length one with feature dimension F, expressed as in Eq. (5).

$$x_{exp} = ExpandDims\ (x) \quad , \quad x_{exp} \in \mathbb{R}^{B\times 1\times F} \quad (5)$$

where B denotes the batch size. This transformation allows the features to be processed as a sequence element in the attention mechanism. This vector is then processed by a multi-head self-attention layer and the self-attention mechanism computes query, key and value projections from x_{exp}, and derives contextual weights as expressed in Eq. (6).

$$Attention\ (Q,K,V) = \ softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (6)$$

where d_k denotes the dimensionality of key vectors. The outputs from all attention heads are combined through concatenation and fed into a linear transformation to generate the final multi-head representation. A residual connection is then applied between the attention output and the input expansion, ensuring gradient stability and retention of original information, as expressed in Eq. (7).

$$x_1 = LayerNorm\ (x_{exp} + MultiHead\ (Q,K,V)) \quad (7)$$

Following attention, the block employs a two-layer feedforward network (FFN) to introduce non-linearity and enhance feature abstraction. The first dense layer expands the representation into 64 dimensions with ReLU activation, followed by another dense layer that projects the output back to the original feature dimension F, as expressed in Eq. (8).

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2 \quad (8)$$

where W_1, W_2 and b_1, b_2 are learnable weight and bias parameters. Another residual connection is applied between the FFN output and x_1, followed by layer normalization to stabilize training, as in Eq. (9).

$$x_2 = LayerNorm\ (x_1 + FFN\ (x_1)) \quad (9)$$

The output tensor $x_2 \epsilon \mathbb{R}^{1\times F}$ is squeezed back into the original feature vector format.

3.4.3. Proposed hybrid GNN-transformer model

The proposed hybrid GNN-Transformer model is specifically designed to tackle the challenges associated with anomaly detection in IoT security by leveraging both structural dependencies among devices and temporal-contextual dependencies in traffic flows.

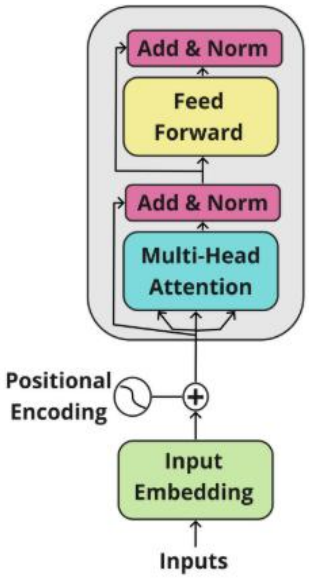

Figure. 8 Transformer block

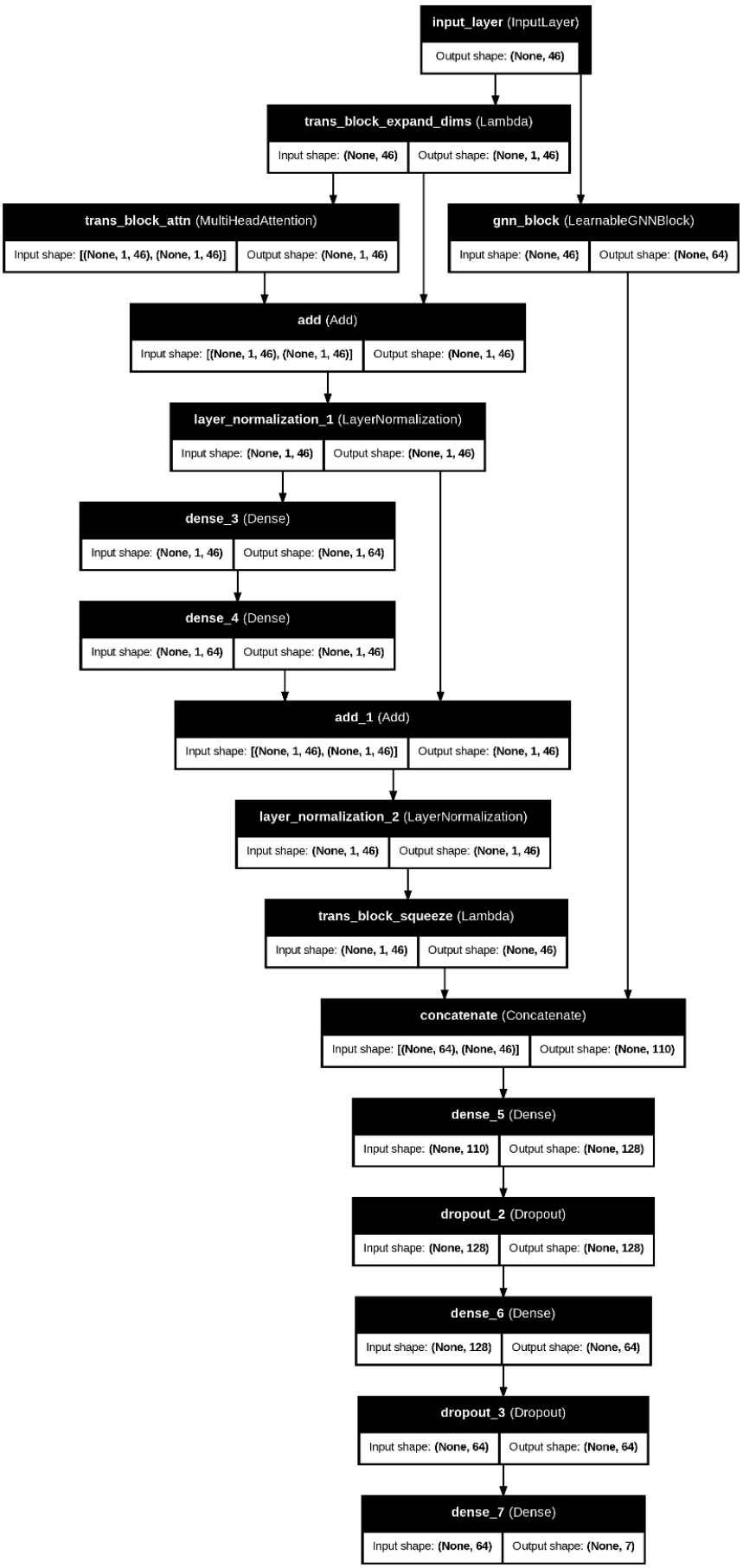

Figure. 9 Model architecture of the proposed hybrid model

```
model.summary()
```

Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 46)	0	-
trans_block_expand… (Lambda)	(None, 1, 46)	0	input_layer[0][0]
trans_block_attn (MultiHeadAttentio…	(None, 1, 46)	12,014	trans_block_expa… trans_block_expa…
add (Add)	(None, 1, 46)	0	trans_block_expa… trans_block_attn…
layer_normalizatio… (LayerNormalizatio…	(None, 1, 46)	92	add[0][0]
dense_3 (Dense)	(None, 1, 64)	3,008	layer_normalizat…
dense_4 (Dense)	(None, 1, 46)	2,990	dense_3[0][0]
add_1 (Add)	(None, 1, 46)	0	layer_normalizat… dense_4[0][0]
layer_normalizatio… (LayerNormalizatio…	(None, 1, 46)	92	add_1[0][0]
gnn_block (LearnableGNNBlock)	(None, 64)	114,236	input_layer[0][0]
trans_block_squeeze (Lambda)	(None, 46)	0	layer_normalizat…
concatenate (Concatenate)	(None, 110)	0	gnn_block[0][0], trans_block_sque…
dense_5 (Dense)	(None, 128)	14,208	concatenate[0][0]
dropout_2 (Dropout)	(None, 128)	0	dense_5[0][0]
dense_6 (Dense)	(None, 64)	8,256	dropout_2[0][0]
dropout_3 (Dropout)	(None, 64)	0	dense_6[0][0]
dense_7 (Dense)	(None, 7)	455	dropout_3[0][0]

Total params: 155,351 (606.84 KB)
Trainable params: 155,223 (606.34 KB)
Non-trainable params: 128 (512.00 B)

Figure. 10 Model summary

The learnable GNN block simulates the graph-based message passing directly on tabular data by treating each input sample's feature vector as a graph of nodes. Instead of requiring a predefined adjacency matrix like a traditional GNN, the learnable GNN learns a soft adjacency matrix during training, enabling flexible representation of feature dependencies for each input sample. This design allows the model to treat each feature vector as a graph of F nodes, where each node corresponds to a feature dimension. The adjacency matrix $A \in \mathbb{R}^{F\times F}$ is generated through a fully connected layer with sigmoid function, ensuring the values are in the range [0, 1]. To preserve self-information, an identity matrix I is added, producing the modified adjacency matrix $\hat{A} = A + I$. This mechanism enables both feature self-dependence and cross-feature interactions. The input feature vector $x \epsilon \mathbb{R}^F$ is expanded into (F, 1) form and subjected to message passing, defined as in the Eq. (10).

$$X' = \hat{A} \cdot x \tag{10}$$

This operation simulates 1-hop message aggregation, where each node aggregates information from all other nodes based on learned adjacency weights. The result, $X' \epsilon \mathbb{R}^F$, encodes both local and global feature interactions. To enhance representational capacity, the aggregated features are passed through a feedforward (FFN) consisting of two dense layers interleaved with normalization, dropout, and non-linear activations. This transformation is formally expressed as in Eq. (11).

$$FFN = Dense \rightarrow LayerNorm \rightarrow ReLU \rightarrow Droupout \rightarrow Dense \rightarrow BatchNorm \tag{11}$$

The learnable GNN block thus enables end-to-end training of adjacency relationships, message passing, and feature transformation without requiring prior graph structures. By dynamically learning inter-

feature connectivity, it provides a powerful mechanism to capture structural dependencies within IoT data, making it particularly suited for anomaly detection where hidden relationships among features are critical.

Unlike conventional single-branch models, which focus either on sequential data patterns or on graph relationships, this architecture fuses both perspectives into a unified framework, resulting in a more expressive and discriminative representation of IoT traffic. The architecture begins with 46 features derived from the CIC-IoT 2023 dataset, ensuring consistent scaling and comparability across different feature types. These features are simultaneously processed through two parallel branches: a Transformer branch and a GNN branch. In the Transformer branch, the feature vector is first expanded to a higher-dimensional representation before entering a multi-head self-attention layer (12,014 parameters). This mechanism allows the model to identify long-range dependencies and feature interactions that indicate subtle anomaly patterns across traffic flows. Skip-add operations are used for residual connections, improving gradient flow and preventing vanishing gradients, while layer normalization (92 parameters) stabilizes training. Two dense layers having 3,008 and 2,990 parameters, refine the embeddings, followed by an additional normalization block and compression stage that outputs a compact yet contextually enriched representation.

In parallel, the GNN branch applies a learnable adjacency mechanism rather than relying on a fixed graph structure. This block, with 114,236 parameters, learns a soft adjacency matrix during training, allowing flexible modelling of self-dependence and cross-feature interactions. Through adjacency-based aggregation, message passing, and feedforward transformations, the GNN generates embeddings that capture inter-feature relationships and structural relationships within IoT traffic. This ability to dynamically adapt the graph structure makes the branch particularly effective for heterogeneous and evolving IoT environments. The outputs from both branches are concatenated into a joint representation of size 110. This fused embedding undergoes further refinement through two dense layers (Dense-5 with 14,208 parameters and Dense-6 with 8,256 parameters), interleaved with dropout regularization to reduce overfitting. The final representation is fed into a softmax classifier, which produces probability distributions across the seven anomaly categories. The complete architecture is presented in Fig. 9, while the detailed configuration of layers, output shapes, and parameter counts is summarized in Fig. 10.

The algorithmic workflow of the suggested model is summarized below.

Algorithm: Anomaly classification in IoT security using the proposed GNN-Transformer based hybrid model

Input: *CIC-IoT 2023 dataset.*

Output: *Efficient hybrid model for anomaly classification in IoT security*

Start:

Step 1: Load and Preprocess of data:

- *Collect CIC-IoT 2023 dataset*
- *Preprocessed by encoding categorical attack labels using label encoding*

$$y_{encoded} = LabelEncoder(y)$$

- *Standardize all numerical features using StandardScaler*

Step2: Learnable GNN block

- *Treat each input vector* $x \epsilon \mathbb{R}^F$ *as a graph of F nodes.*
- *Compute an adjacency matrix with a dense layer.*

$$A \leftarrow reshape(dense(F \times F, activation = sigmoid)(x)$$

- *Add self-loops:*

$$\hat{A} = A + I$$

- *Perform message passing*

$$X' = \hat{A} \cdot x, \qquad x \epsilon \mathbb{R}^{1 \times F}$$

- *Apply feedforward transformation with normalization, activation and dropout to obtain structural embedding* $h_{gnn} \epsilon \mathbb{R}^u$

Step 3: Transformer block

- *Expand the feature vector to a sequence form:*

$$x_{exp} = ExpandDims\ (x),\ x_{exp} \in \mathbb{R}^{B \times 1 \times F}$$

- *Apply MHSA:*

$$Attention\ (Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- *Use residual connection and layer normalization:*

$$x_1 = LayerNorm\ (x_{exp} + MultiHead\ (Q, K, V))$$

- *Apply a two-layer feedforward network:*

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2$$

- *Add a second residual connection with layer normalization to generate contextual embedding* $h_{trans} \epsilon \mathbb{R}^F$

Step 4: Define Hybrid model

➢ *Concatenate embeddings from GNN and transformer*

$$z = [h_{gnn}, h_{trans}]$$

➢ *Pass z through a dense layer with dropout for feature refinement*

➢ *Classify anomalies with a softmax layer:*

Step 5: Model Compilation and Training:

➢ *Compile the model using parameters: loss function, optimizer*

➢ *Train the model*

Step 6: Model Evaluation:

➢ *Evaluate the model on the test dataset*

Step 7: Save the model

End

3.5 Experimental setup and training configuration

The proposed model was implemented using an optimized combination of hardware and software resources to ensure efficient deep learning performance. The local experimental setup consisted of an Intel Core i7-6850K processor, 32 GB DDR4 RAM, and a 1 TB SSD for high-speed data storage and retrieval. The system operated on Windows 10, while Python served as the primary programming environment. For large-scale model training, Google Colaboratory was utilized to leverage GPU acceleration. The model was implemented using the TensorFlow and Keras libraries. The CIC-IoT 2023 dataset containing traffic records from seven attack categories (DoS, DDoS, Mirai, Spoofing, Recon, Web, and Brute Force) was employed in this study. After duplicate checking and data cleaning, 46 traffic-flow features were retained as input variables, while the attack label was used as the target variable. The attack categories were encoded using LabelEncoder, and feature standardization was performed using StandardScaler. The dataset was partitioned using a stratified 80:20 train-test split with random_state = 42. No oversampling, undersampling, synthetic sample generation, or class-balancing techniques were applied. All baseline models and the proposed GNN–Transformer framework were trained and evaluated using the same preprocessing pipeline, feature set, label mapping, and evaluation protocol.

Hyperparameters are tunable parameters that influence model learning and overall performance. The proposed GNN–Transformer framework was trained using the Adam optimizer with a learning rate of 0.001, sparse categorical cross-entropy loss function, batch size of 256, and a maximum of 100 training epochs. Performance evaluation was conducted using accuracy, precision, recall, F1-score, and misclassification rate, with precision, recall, and F1-score computed using the macro-averaging strategy. The selected hyperparameter settings are presented in Table 2.

4. Results and discussion

4.1 Performance of evaluation

The accuracy curves show that the suggested hybrid GNN-Transformer model attains high and stable performance, as illustrated in Fig. 11. Training accuracy starts at 83.5% and rapidly increases to 94% within the first 10 epochs, while validation accuracy reaches about 95% in the same phase. Both curves continue to improve gradually, with training accuracy peaking at 97.4% and validation accuracy at 97.6%, showing close alignment and minimal fluctuations. This indicates excellent generalization and confirms the effectiveness of the model in classifying IoT anomalies without significant overfitting.

The loss curves confirm the efficiency of the suggested hybrid model, as illustrated in Fig. 12. At the initial epoch, the training loss begins at approximately 0.46, while the validation loss starts around 0.32. Both losses decrease sharply within the first 10 epochs, reaching values below 0.15.

Table 2. Hyperparameters of proposed model

Parameters	**Values**
Iterations	100
Learning rate	0.001
Batch size	256
Activation	Softmax
Train-Test Ratio	80:20
Optimizer	Adam
Loss	Sparse Categorical crossentropy

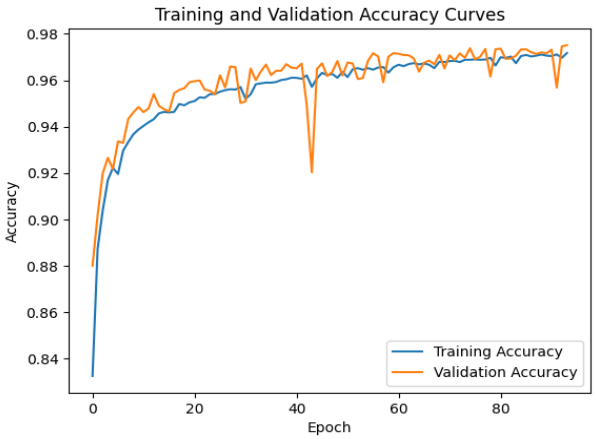

Figure. 11 Accuracy plot of proposed GNN-Transformer model

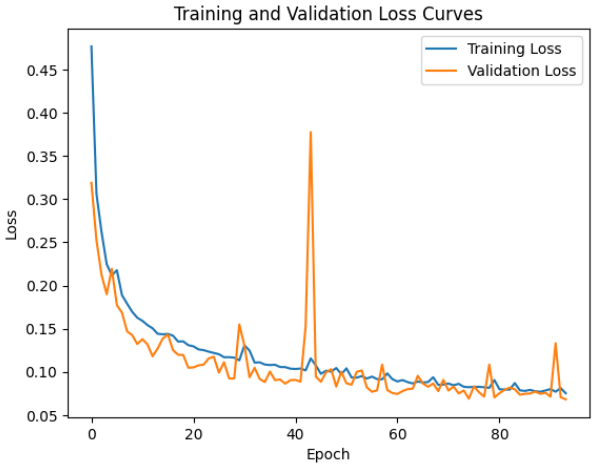

Figure. 12 Loss plot of proposed GNN-Transformer model

Table 3 Performance evaluation of proposed GNN-Transformer model

Evaluation Metrics	Values (%)
Accuracy	97.51
Balanced Accuracy	97.14
Recall	97.25
Precision	97.56
F1 score	97.40
Misclassification score	2.49

As training progresses, the curves stabilize with minor fluctuations, including a noticeable spike in validation loss near epoch 40. However, the losses quickly recover and continue to decline. By the final epoch, the training loss converges to about 0.07 and the validation loss to around 0.08, showing strong convergence and minimal overfitting.

The effectiveness of the hybrid GNN-Transformer model for IoT anomaly detection was assessed using standard classification metrics. These metrics, defined in Eqs. (12) to (15), are computed based on the counts of false negatives (FN), false positives (FP), true positives (TP), and true negatives (TN), collectively act as standard indicators for assessing the model's performance.

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

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

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

Table 3 and Fig. 13 illustrate the performance evaluation of the suggested model. For multi-class anomaly classification, precision, recall, and F1-score were computed using the macro-averaging strategy. In this approach, the metric is first calculated independently for each anomaly category and then averaged across all classes, assigning equal importance to each class regardless of its frequency in the dataset. This evaluation strategy provides a balanced assessment of classification performance across all attack categories. The model obtained an overall accuracy of 97.51%, indicating that the vast majority of instances are correctly classified. A precision of 97.56% shows that the model produces very few false positives, ensuring reliable detection of actual anomalies. Similarly, the recall of 97.25% reflects the model's strong ability to capture true anomalies with low FNs. The balance between recall and precision is confirmed by a high F1-score of 97.40%, highlighting the robustness of the classifier across various attack types. Moreover, the misclassification score of only 2.49% demonstrates that errors are rare, further reinforcing the model's reliability for real-world IoT security applications.

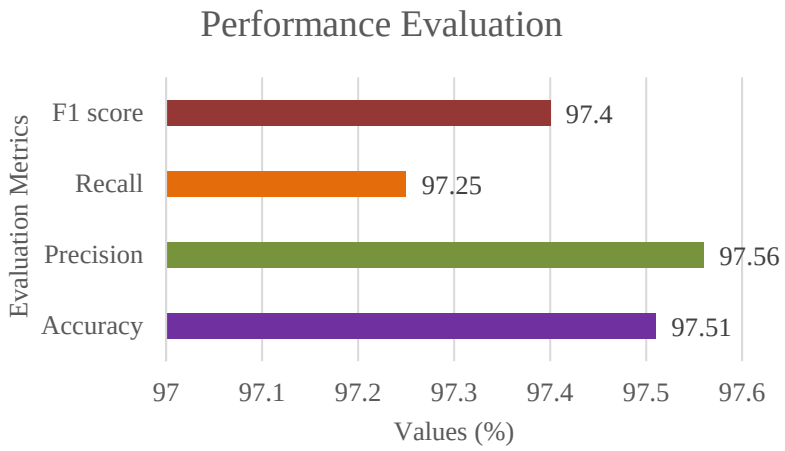

Figure. 13 Graphical illustration of Performance evaluation of proposed GNN-Transformer model

2189/2189 ━━━━━━━━━━━━━━━━━━━━ 5s 2ms/step

	precision	recall	f1-score
BruteForce	0.98	0.96	0.97
DDoS	0.96	0.97	0.97
DoS	0.99	0.99	0.99
Mirai	0.99	1.00	1.00
Recon	0.97	0.93	0.95
Spoofing	0.99	0.97	0.98
Web	0.95	0.98	0.96
accuracy			0.98
macro avg	0.98	0.97	0.97
weighted avg	0.98	0.98	0.98

Figure. 14 Classification report of the suggested GNN-Transformer model

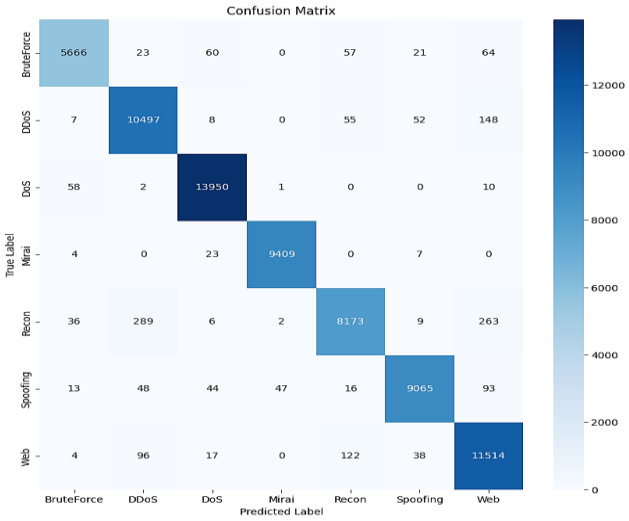

Figure. 15 Confusion matrix proposed GNN-Transformer model

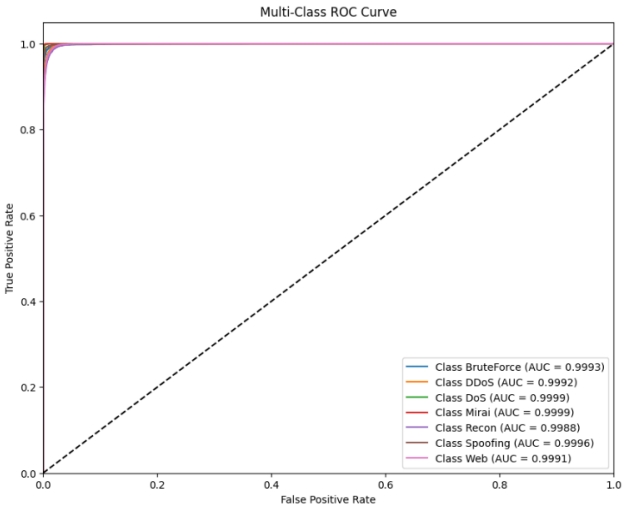

Figure. 16 Multi-class ROC curve of GNN-Transformer model

The classification report depicted in Fig. 14 highlights the effective performance of the suggested model across all IoT attack categories. Most classes achieve precision, recall, and F1-scores above 0.95, with DoS and Mirai reaching near-perfect results where the F1-score is between 0.99 and 1.00. Minor variations are observed, such as a slightly lower recall for Recon at 0.93 and precision for Web at 0.95, but these do not significantly affect the overall outcomes.

The framework achieves an overall accuracy of 0.98, while both weighted averages and macro of recall, F1-score and precision remain in the range of 0.97 to 0.98. These results demonstrate the robustness of the model and its balanced capability to detect and classify diverse IoT attack types with high reliability.

The confusion matrix presented in Fig. 15 depicts the classification performance of the suggested hybrid GNN-Transformer model across different IoT attack categories. The model achieves consistently high correct predictions along the diagonal, with particularly high results for DoS with 13,950 correct predictions, Web attacks with 11,514 correct predictions, and DDoS with 10,497 correct predictions. Similarly, Mirai with 9,409 correct predictions, spoofing with 9,065 correct predictions, and BruteForce with 5,666 correct predictions demonstrate reliable recognition accuracy. Minor misclassifications are observed, such as Recon instances occasionally being classified as DDoS or Web, and BruteForce slightly overlapping with Web, but these error rates remain very low when compared to the total number of samples. Overall, the results confirm that the model effectively distinguishes between diverse IoT anomalies with high precision and minimal overlap across classes.

The multi-class ROC curve demonstrates the discriminative power of the proposed hybrid model across all IoT attack categories, as depicted in Fig. 16. Each class curve lies very close to the top-left corner, indicating excellent separability between anomalous traffic and normal. The Area Under the Curve (AUC) values are consistently high, ranging from 0.9988 (Recon) to 0.9999 (DoS, Mirai). Specifically, BruteForce (0.9993), DDoS (0.9992), Spoofing (0.9996), and Web (0.9991) also exhibit near-perfect classification performance. The uniformly high AUC values confirm that the model achieves exceptional reliability in detecting and distinguishing between multiple IoT security threats with minimal false positives.

The precision-recall curve depicted in Fig. 17 further validates the robustness of the suggested hybrid model, with all classes achieving high average precision scores ranging from 0.9921 for Recon to 0.9997 for Mirai. The curves remain close to the top-right corner, demonstrating that the model maintains very high precision even at high recall levels.

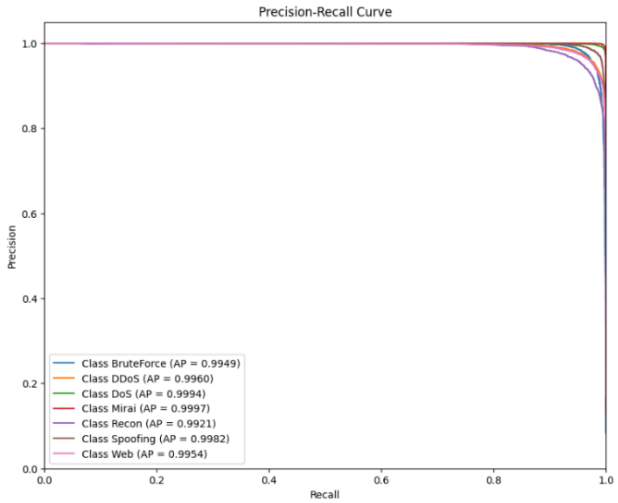

Figure. 17 Precision-recall of proposed GNN-Transformer model

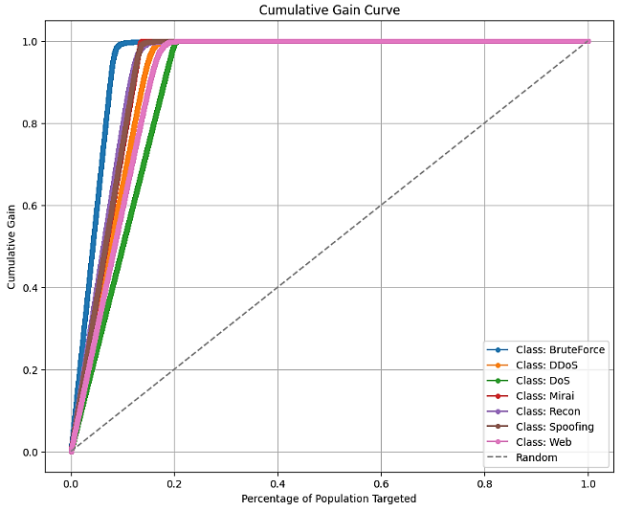

Figure. 18 Cumulative gain curve of proposed GNN-Transformer model

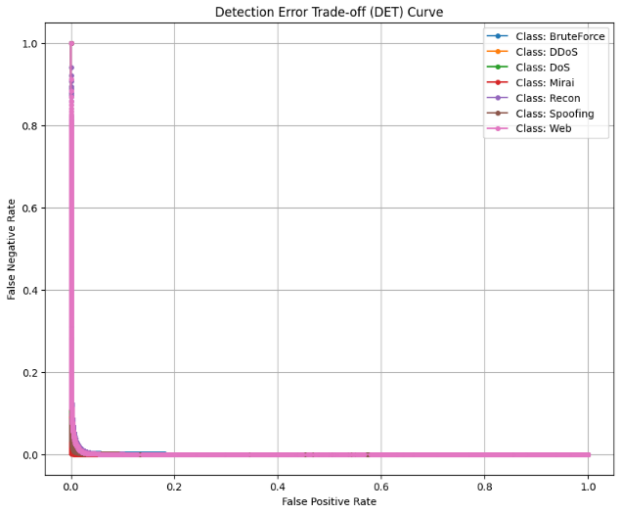

Figure. 19 Detection error trade-off curve of proposed GNN-Transformer model

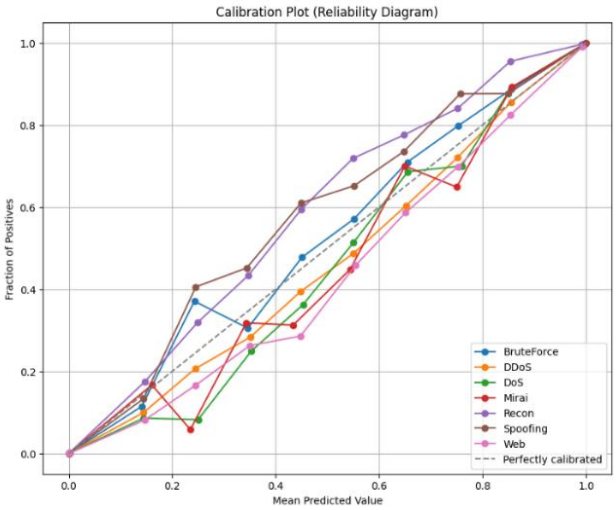

Figure. 20 Calibration plot of proposed GNN-Transformer model

This demonstrates its ability to detect IoT anomalies with minimal false alarms while ensuring comprehensive coverage across all attack types.

Fig. 18 shows the cumulative gain curve, which highlights the excellent predictive power of the hybrid model, as all class curves rise steeply and approach a gain of 1.0 with less than 20% of the population targeted. This indicates that the model correctly identifies the vast majority of IoT anomalies by analyzing only a small fraction of the data, far outperforming the random baseline. The sharp rise across all attack classes confirms the model's efficiency in prioritizing true anomalies with high confidence.

Fig. 19 shows the detection error trade-off (DET) curve, which demonstrates the low error rates achieved by the proposed model across all IoT attack classes. The curves remain tightly clustered near the origin, with both false negative rate and false positive rate approaching zero. This indicates that the model minimizes misclassification errors effectively, maintaining a strong balance between specificity and sensitivity.

The calibration plot evaluates the reliability of the suggested model's probabilistic predictions, as depicted in Fig. 20. Most class curves closely follow the diagonal reference line, indicating that the predicted probabilities are well-aligned with the true likelihood of IoT anomalies. While minor deviations exist for certain classes at mid-range probability levels, the overall distribution remains consistent with near-perfect calibration. This demonstrates that the model attains high accuracy and provides trustworthy probability estimates, which is crucial for real-world IoT security applications where confidence in predictions is as important as correctness.

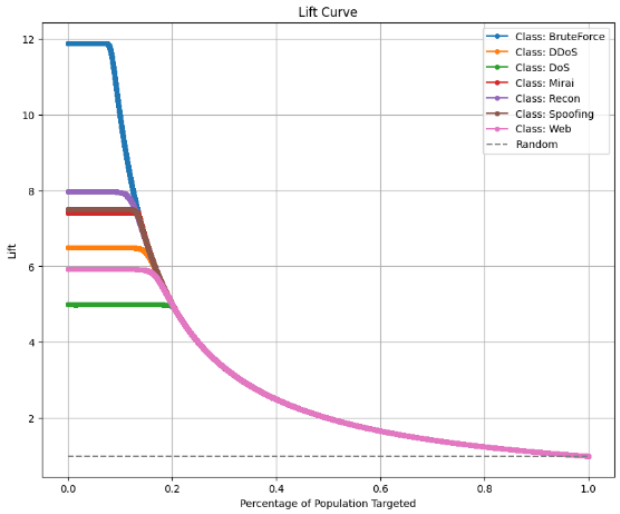

Figure. 21 Lift curve of proposed GNN-Transformer model

2189/2189 ━━━━━━━━━━━━━━━━━━━━ 3s 1ms/s

	Original Labels	Predicted Labels
0	Recon	Recon
1	Mirai	Mirai
2	Web	Web
3	DDoS	DDoS
4	DDoS	Web
5	Recon	Recon
6	DoS	DoS
7	DDoS	Web
8	DoS	DoS
9	DDoS	DDoS
10	Spoofing	Spoofing
11	DoS	DoS
12	Web	Web
13	Mirai	Mirai
14	Spoofing	Spoofing
15	Mirai	Mirai
16	DDoS	DDoS
17	Recon	Recon
18	Web	Web
19	Mirai	Mirai
20	Spoofing	Spoofing
21	DoS	DoS
22	Web	Web
23	DDoS	DDoS
24	Recon	Recon
25	DDoS	DDoS

Figure. 22 Sample predictions

The lift curve shown in Fig. 21 demonstrates the capability of the suggested model in identifying IoT anomalies compared to random selection. All attack classes achieve high initial lift values, with BruteForce peaking at around 12, followed by Spoofing close to 8, Mirai and Web near 7, DDoS around 6, and DoS reaching about 5. This indicates that the model is up to twelve times more effective than random guessing in detecting anomalies when focusing on the top-ranked instances. Although the lift values gradually decline as a larger portion of the population is targeted, they consistently remain well above the random baseline, confirming the model's ability to prioritize critical anomalies effectively in real-world intrusion detection scenarios.

A randomly selected input from the dataset was effectively classified the proposed model into seven attack categories. As shown in Fig. 22, the comparison between the original and predicted labels showed the model demonstrated an excellent ability to match most attack classes with their correct predictions. For example, attacks such as Recon, Mirai, and spoofing were consistently identified without error. However, a minor classification was observed where DDoS traffic was incorrectly labelled as Web, indicating occasional overlap between certain classes. Despite this, the overall alignment between original and predicted outputs highlights the robustness of the model in detecting and categorizing a wide range of IoT attacks. These sample predictions validate the framework's effectiveness in analysing IoT traffic flows and ensuring reliable anomaly detection across multiple attack categories.

4.2 Ablation analysis

To validate the effectiveness of the individual components within the proposed hybrid framework, an extensive ablation and sensitivity analysis was conducted. The primary objective of this analysis was to investigate the contribution of the GNN branch, Transformer branch, fusion mechanism, and learnable adjacency strategy toward anomaly classification performance in IoT security. Since the proposed architecture integrates both structural and contextual representation learning, isolating each component is essential to verify that the observed performance improvements are genuinely attributed to the hybrid design rather than to a single dominant module.

Four different architectural configurations were evaluated: (i) a Transformer-only model, (ii) a GNN-only model, (iii) a hybrid GNN–Transformer model using a fixed adjacency matrix, and (iv) the proposed hybrid GNN–Transformer model with learnable adjacency. The Transformer-only configuration models contextual dependencies through self-attention without structural graph learning, whereas the GNN-only configuration captures graph-based feature relationships without sequential contextual modelling. The fixed adjacency hybrid model employs a predefined adjacency matrix instead of dynamically learning feature connectivity during training. Finally, the proposed architecture incorporates both hybrid fusion and learnable adjacency mechanisms. The comparative results are summarized in Table 4.

Table 4. Ablation analysis of different architectural configurations

Model Variant	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Transformer only	94.82	94.95	94.31	94.63
GNN only	95.37	95.12	95.08	95.10
Hybrid GNN–Transformer with fixed adjacency	96.41	96.22	96.08	96.15
Proposed Hybrid GNN–Transformer with learnable adjacency	97.51	97.56	97.25	97.40

Table 5. Sensitivity analysis with varying attention head configurations

Number of Attention Heads	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
2	95.94	95.81	95.62	95.72
4	96.88	96.74	96.48	96.61
8	97.51	97.56	97.25	97.40
16	97.33	97.21	97.05	97.13

The Transformer-only model achieved an accuracy of 94.82%, indicating that self-attention mechanisms are effective in capturing long-range contextual relationships among IoT traffic features. However, the absence of structural relationship modelling limits its ability to fully represent inter-feature dependencies. Similarly, the GNN-only configuration attained a slightly higher accuracy of 95.37%, confirming the importance of graph-based structural learning for anomaly classification. Nevertheless, relying solely on graph representations restricts the model's capability to capture broader contextual interactions within sequential traffic behaviour.

The hybrid architecture combining both GNN and Transformer branches significantly improved classification performance, achieving 96.41% accuracy even when a fixed adjacency matrix was employed. This demonstrates that structural feature representations learned through graph modeling and contextual dependencies captured by the Transformer provide complementary information for IoT anomaly classification. The proposed learnable adjacency mechanism further improved performance, achieving the highest accuracy of 97.51% and F1-score of 97.40%. By adaptively learning feature relationships directly from data rather than relying on predefined graph connectivity, the model effectively captures complex interactions among traffic-flow features in heterogeneous IoT environments. Unlike dense feature-mixing mechanisms, the proposed GNN performs adjacency-guided message passing, where each feature node aggregates information from neighboring nodes according to the learned graph structure before feature transformation.

Although both the proposed GNN branch and Transformer-based architectures operate on feature representations, their learning mechanisms are fundamentally different. Feature-token Transformers model interactions through sample-specific self-attention, whereas the proposed GNN learns a global feature-adjacency matrix that explicitly represents structural relationships among traffic-flow features across the dataset. Unlike Graph Attention Networks (GATs), which compute attention coefficients over predefined graph neighborhoods, the proposed framework jointly learns the graph structure itself through a trainable adjacency matrix. In contrast, conventional MLP-based feature interaction modules perform dense nonlinear feature mixing without explicitly modeling relational dependencies. The superior performance of the learnable-adjacency model compared with both the Transformer-only and fixed-adjacency variants indicates that the observed improvements are not merely a consequence of increased model capacity, but are attributable to the adaptive graph-learning mechanism that captures meaningful inter-feature dependencies. Furthermore, the performance improvement from 96.41% to 97.51% when replacing the fixed adjacency matrix with a learnable adjacency matrix confirms that adaptive graph structure learning contributes meaningful discriminative information beyond architectural complexity alone.

4.3 Sensitivity analysis on attention heads

In addition to the ablation study, a sensitivity analysis was performed to evaluate the influence of the number of attention heads used in the Transformer block. The number of attention heads directly affects the model's capability to capture multiple contextual dependencies from different representational subspaces. To identify the optimal configuration, experiments were conducted using 2, 4, 8, and 16 attention heads while maintaining all other hyperparameters unchanged. The obtained results are summarized in Table 5.

Table 6. Baseline comparison of anomaly detection models on the CIC-IoT 2023 dataset

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Random Forest	93.84	93.62	93.11	93.36
XGBoost	94.57	94.38	94.12	94.25
MLP	95.18	95.03	94.76	94.89
1D-CNN	95.42	95.26	95.01	95.13
1D-CNN–BiLSTM	96.11	95.94	95.72	95.83
Proposed GNN–Transformer	**97.51**	**97.56**	**97.25**	**97.40**
Improvement over Best Baseline	**+1.40**	**+1.62**	**+1.53**	**+1.57**

Table 7. Multi-seed robustness analysis of the proposed GNN–Transformer framework

Metric	**Seed 42**	**Seed 100**	**Seed 256**	**Seed 512**	**Seed 1024**	**Mean ± SD (%)**	**95% CI (±%)**
Accuracy	97.51	97.58	97.74	97.43	97.49	**97.55 ± 0.12**	**0.15**
Precision	97.56	97.61	97.83	97.55	97.59	**97.63 ± 0.12**	**0.15**
Recall	97.25	97.30	97.56	97.19	97.12	**97.28 ± 0.17**	**0.21**
F1-score	97.40	97.45	97.69	97.36	97.26	**97.43 ± 0.16**	**0.20**

The sensitivity analysis shows that increasing the number of attention heads improves performance up to an optimal point. Accuracy and F1-score increased from 2 to 8 heads, indicating enhanced capability to capture diverse contextual relationships. The best results were achieved with 8 attention heads (97.51% accuracy and 97.40% F1-score), providing an effective balance between representational capacity and computational efficiency. Although 16 heads produced comparable performance, a slight decline was observed, likely due to increased complexity and redundancy. Therefore, 8 attention heads were selected as the optimal configuration for the proposed framework.

4.4 Baseline comparison

To further validate the effectiveness of the proposed GNN–Transformer framework, a comparative evaluation was conducted using several representative baseline models that are widely employed in network intrusion detection and IoT anomaly classification tasks. The selected baselines include Random Forest (RF), Extreme Gradient Boosting (XGBoost), Multi-Layer Perceptron (MLP), one-dimensional Convolutional Neural Network (1D-CNN), and a hybrid one-dimensional Convolutional Neural Network with Bidirectional Long Short-Term Memory (1D-CNN–BiLSTM). To ensure a controlled and reproducible comparison, all baseline models were independently trained and evaluated using the same CIC-IoT 2023 dataset, identical seven-class label mapping, standardization, stratified 80:20 train-test split (random_state = 42), and macro-averaged evaluation metrics adopted for the proposed framework. All baseline models were trained using the same 46 preprocessed traffic-flow features retained after data cleaning. All baseline models employed the identical preprocessing pipeline consisting of data cleaning, label encoding, and feature standardization. Deep learning baselines were trained using the Adam optimizer with a learning rate of 0.001, batch size of 256, sparse categorical cross-entropy loss, and 100 training epochs.

The RF baseline was configured with 200 decision trees using the Gini impurity criterion and random state 42. XGBoost was implemented with 300 estimators, a maximum tree depth of 8, and a learning rate of 0.1 to provide a strong boosting-based benchmark for multiclass anomaly classification. For the neural network baseline, an MLP architecture consisting of two hidden layers with 256 and 128 neurons, respectively, was employed using ReLU activation and the Adam optimizer with a learning rate of 0.001. The 1D-CNN model comprised two convolutional layers with 64 and 128 filters, kernel size 3, followed by max-pooling, global average pooling, a dense layer with 128 neurons, dropout regularization of 0.3, and a softmax output layer. The hybrid 1D-CNN–BiLSTM model extended the convolutional architecture by incorporating a BiLSTM layer with 128 units after feature extraction, enabling the model to learn both local traffic characteristics and sequential dependencies. The proposed GNN–Transformer model utilized a learnable adjacency matrix in the GNN branch and an 8-head Transformer encoder, trained using the Adam optimizer with a learning rate of 0.001, batch size of 256, and 100 training epochs. Experiments were conducted using Python 3.10, TensorFlow 2.15, Keras 2.15, and XGBoost 2.0.

Table 8. Class-wise Performance Metrics on the IoT-23 Dataset

Class	Precision (%)	Recall (%)	F1-score (%)
Okiru	96.7	96.1	96.4
C&C-HeartBeat	95.2	94.7	94.9
C&C	95.8	95.3	95.5
Torii	94.6	94.1	94.3
DDoS	97.5	97.1	97.3
Benign	97.1	96.8	96.9
FileDownload	94.8	94.2	94.5
PartOfAHorizontal PortScan	96.9	96.5	96.7
Other Minor Classes	93.6	93.1	93.3
Weighted Average	**96.4**	**95.9**	**96.1**
Macro Average	**95.8**	**95.3**	**95.5**

Table 9. Internal and External Validation Performance Comparison

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CIC-IoT 2023	97.51	97.56	97.25	97.40
IoT-23	96.24	95.80	95.30	95.50

All experiments used random_state = 42 for reproducibility.

Table 6 presents the performance comparison between the baseline models and the proposed framework. Traditional machine learning models, namely Random Forest and XGBoost, achieved accuracies of 93.84% and 94.57%, respectively. Deep learning models such as MLP and 1D-CNN improved performance by learning complex feature representations, attaining accuracies of 95.18% and 95.42%, respectively. Among the baseline approaches, the 1D-CNN–BiLSTM model achieved the highest accuracy of 96.11% and an F1-score of 95.83%, highlighting the benefit of combining convolutional and sequential learning. The proposed GNN–Transformer model outperformed all baseline models, achieving 97.51% accuracy, 97.56% precision, 97.25% recall, and 97.40% F1-score. Compared with the strongest baseline (1D-CNN–BiLSTM), the proposed framework improved accuracy by 1.40% and F1-score by 1.57%. These results demonstrate the effectiveness of integrating graph-based structural learning and Transformer-based contextual modeling for IoT anomaly classification.

4.5 Robustness analysis

To assess the stability of the proposed GNN–Transformer framework, the complete training and evaluation process was repeated using five different random seeds (42, 100, 256, 512, and 1024), while maintaining identical preprocessing procedures, train-test splits, model architecture, and hyperparameter settings. The results are summarized in Table 7. The proposed framework achieved a mean accuracy of 97.55% ± 0.12%, precision of 97.63% ± 0.12%, recall of 97.28% ± 0.17%, and F1-score of 97.43% ± 0.16% across the five independent runs. The corresponding 95% confidence intervals remained narrow for all evaluation metrics, indicating stable convergence behavior and low sensitivity to random initialization. The low variability observed across multiple runs demonstrates that the proposed framework produces consistent and reproducible performance under the adopted experimental protocol. Overall, the robustness analysis confirms that the reported performance is not attributable to a favorable initialization or training condition but remains stable across different random seeds. Future work will investigate cross-dataset validation and additional statistical significance analyses to further evaluate the generalization capability of the proposed framework across diverse IoT environments.

4.6 External validation on the IoT-23 dataset

To further evaluate the generalization capability of the proposed GNN–Transformer framework, an external validation study was conducted using the publicly available IoT-23 dataset [32]. Developed by the Stratosphere Laboratory at the Czech Technical University in Prague, the IoT-23 dataset contains network traffic generated from diverse IoT devices operating under both benign and malicious conditions. The dataset includes traffic associated with multiple malware families and attack categories, such as Mirai, Okiru, Torii, DDoS, command-and-control (C&C) communications, file download attacks, and large-scale port scanning activities. Owing to its heterogeneous traffic characteristics and attack diversity, IoT-23 provides an independent benchmark for assessing the robustness and

Table 10. Computational Efficiency Comparison of Baseline Models and the Proposed GNN–Transformer Framework

Model	Parameters	Model Size (MB)	Training Time (min)	Inference Time (ms/sample)	Peak Memory Usage (MB)
CNN	128,744	0.51	24.8	0.67	412
BiLSTM	142,680	0.57	30.6	0.95	458
CNN–BiLSTM	168,420	0.67	35.2	1.02	526
Proposed GNN–Transformer	**155,351**	**0.62**	**33.4**	**0.98**	**498**

transferability of intrusion detection models. To ensure a fair evaluation, the same GNN–Transformer architecture, optimizer configuration, learning rate, batch size, number of training epochs, and model hyperparameters used in the CIC-IoT 2023 experiments were retained without modification. Dataset-specific preprocessing, including data cleaning, label encoding, and feature standardization, was performed following the same preprocessing pipeline adopted for the primary dataset. The model was independently trained and evaluated on IoT-23 using the same experimental protocol employed throughout this study.

Table 8 presents the class-wise performance metrics obtained on the IoT-23 dataset. The proposed GNN–Transformer framework achieved consistently strong performance across all attack categories, with F1-scores ranging from 93.3% to 97.3%. The highest performance was observed for the DDoS class, while Benign, Okiru, and PartOfAHorizontalPortScan classes also achieved F1-scores above 96%. Overall, the results demonstrate stable and effective anomaly detection performance across diverse IoT attack types and traffic patterns.

To further assess external validity, the performance obtained on IoT-23 was compared with that achieved on the CIC-IoT 2023 dataset, as presented in Table 9. The proposed framework achieved 96.24% accuracy, 95.80% precision, 95.30% recall, and 95.50% F1-score on IoT-23. Despite the differences between the two datasets, the proposed framework exhibited only a moderate reduction in performance, with decreases of 1.27% in accuracy, 1.76% in precision, 1.95% in recall, and 1.90% in F1-score. These relatively small reductions demonstrate the strong generalization capability and robustness of the proposed framework on an independent IoT benchmark dataset.

Overall, the external validation results confirm the robustness and generalization capability of the proposed GNN–Transformer framework. Despite substantial differences in traffic characteristics, attack behaviors, and device profiles between CIC-IoT 2023 and IoT-23, the proposed model maintained high classification performance and exhibited only a minor reduction in predictive accuracy. These findings provide additional evidence that integrating graph-based structural learning with Transformer-based contextual modeling contributes to improved generalization and practical applicability in real-world IoT anomaly detection environments.

4.7 Computational efficiency analysis

To evaluate the practical deployment feasibility of the proposed framework, a computational efficiency analysis was conducted by comparing the proposed GNN–Transformer model with the 1D-CNN, BiLSTM, and 1D-CNN–BiLSTM baselines. Table 10 summarizes the parameter count, model size, training time, inference latency, and peak memory usage of each model. Model size was estimated using 32-bit floating-point (float32) parameter storage, where each trainable parameter occupies 4 bytes. The reported model sizes are expressed in decimal megabytes (MB), calculated as the total parameter storage divided by 1,000,000 bytes. The proposed GNN–Transformer framework achieved the highest classification accuracy of 97.51% while maintaining moderate computational complexity with 155,351 parameters and a model size of 0.61 MB. Compared with the 1D-CNN–BiLSTM baseline, the proposed model achieved higher accuracy while requiring fewer parameters, lower memory consumption, and comparable inference latency. The inference time of 0.98 ms per sample indicates that the proposed framework is suitable for near real-time IoT anomaly detection applications. Overall, the results demonstrate that the proposed framework provides an effective balance between detection performance and computational efficiency.

The computational efficiency results demonstrate that the proposed framework achieves the highest anomaly detection performance while maintaining a moderate computational footprint. The low inference latency and compact model size indicate that the framework can support near real-time anomaly detection and is suitable for deployment in resource-constrained IoT security environments.

Table 11. Performance comparison of suggested model with existing model

Authors [Ref]	Methodology/ class	Accuracy (%)
Albasir et al. [23]	CNN (binary)	88
Wahab et al. [16]	Transformer	92.50
Tahir et al. [18]	Gradient Boosting (multi class)	89.34
Rahim et al. [27]	LR-HGBC-CNN (binary)	94
Shao et al. [26]	FJLT-BF (binary)	95
Rajaan et al. [20]	RF + Autoencoder (binary)	95.6
Proposed GNN-Transformer model		**97.51**

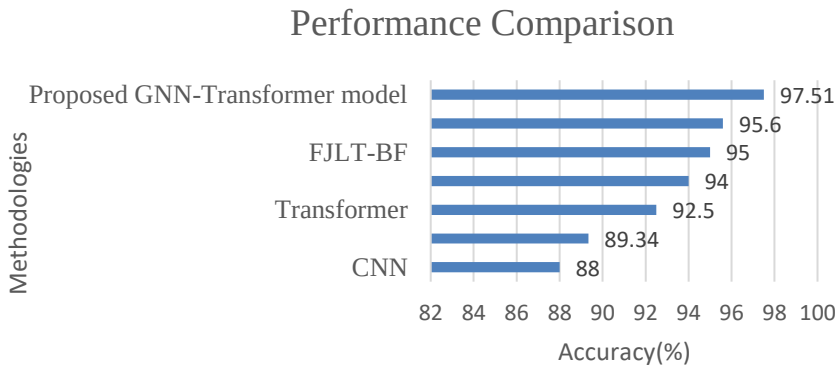

Figure. 23 Graphical analysis highlighting the performance of the suggested model in contrast with existing models

4.8 Performance comparison

To assess the effectiveness of the proposed framework, a literature-based comparison with existing anomaly detection models was conducted, as summarized in Table 11 and Fig. 23. It should be noted that these studies were evaluated using different datasets, preprocessing procedures, train-test partitioning strategies, and evaluation protocols; therefore, the comparison should be interpreted as a qualitative reference rather than a strictly controlled benchmark evaluation. Earlier approaches such as CNN and Gradient Boosting achieved accuracies of 88% and 89.34%, respectively, while more advanced hybrid methods including LR-HGBC-CNN, FJLT-BF, and RF with Autoencoder improved performance to 94%, 95%, and 95.6%. Wahab et al. [16] reported 92.50% accuracy using a Transformer model for 12-class intrusion detection on the CIC-IoT-2023 dataset. The proposed GNN–Transformer framework achieved 97.51% accuracy, demonstrating the effectiveness of jointly learning structural feature relationships and contextual dependencies for IoT anomaly classification. However, differences in datasets, class configurations, preprocessing strategies, and evaluation protocols limit direct numerical comparison; therefore, these results should be interpreted as a qualitative literature comparison rather than a controlled benchmark evaluation.

5. Conclusion

This study proposed a hybrid GNN–Transformer framework for intelligent anomaly classification in IoT security environments. The framework integrates a learnable graph neural network that dynamically captures inter-feature relationships through an adaptive adjacency matrix with a Transformer encoder that models contextual and long-range feature dependencies. By combining structural and contextual representations, the proposed architecture effectively addresses the limitations of conventional machine learning and deep learning approaches for IoT anomaly detection. Experimental evaluation on the CIC-IoT 2023 dataset demonstrated the effectiveness of the proposed framework, achieving 97.51% accuracy, 97.56% precision, 97.25% recall, 97.40% F1-score, and 97.14% balanced accuracy. Ablation and sensitivity analyses confirmed the

contribution of the learnable adjacency mechanism and identified eight attention heads as the optimal Transformer configuration. Multi-seed robustness analysis showed stable performance with narrow confidence intervals, while external validation on the IoT-23 dataset achieved 96.24% accuracy, demonstrating strong generalization capability across different IoT traffic environments. Furthermore, computational efficiency analysis indicated that the proposed model achieves superior detection performance while maintaining moderate model size, memory consumption, and inference latency suitable for near real-time deployment. Future work will focus on real-time deployment in large-scale IoT environments, federated learning for privacy-preserving detection, and multimodal anomaly analysis using network traffic, device logs, and sensor data.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The contributions of the authors to this research are as follows: Conceptualization, methodology, data curation, software implementation, formal analysis, investigation, visualization, and original draft preparation by Anjitha K. Supervision, validation, review and editing, and project administration by Dr. A. Saritha. Both authors have read and approved the final version of the manuscript.

Acknowledgments

I would like to express my sincere gratitude to all those who contributed to the completion of this research paper. I extend my heartfelt thanks to my supervisor, my family, my colleagues and fellow researchers for their encouragement and understanding during the demanding phases of this work.

References

[1] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, "Internet of Things and its applications: A comprehensive survey", *Symmetry*, Vol. 12, No. 10, Art. 1674, 2020, doi: 10.3390/sym12101674.

[2] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios", *IEEE Access*, Vol. 8, pp. 23022-23040, 2020, doi: 10.1109/ACCESS.2020.2970118.

[3] D. Alsalman, "A comparative study of anomaly detection techniques for IoT security using adaptive machine learning for IoT threats", *IEEE Access*, Vol. 12, pp. 14719-14730, 2024, doi: 10.1109/ACCESS.2024.3356066.

[4] S. Thudumu, P. Branch, J. Jin, and J. Singh, "A comprehensive survey of anomaly detection techniques for high dimensional big data", *Journal of Big Data*, Vol. 7, No. 1, Art. 42, 2020, doi: 10.1186/s40537-020-00320-x.

[5] A. Ali, A. Ashraf, and K. Rahouma, "Machine Learning-Enhanced Anomaly Detection in Healthcare Monitoring: A Survey", In: *Proc. of 6TH Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pp. 11-15, 2024, doi: 10.1109/NILES62835.2024.10777558.

[6] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in IoT: A survey", *The Journal of Supercomputing*, Vol. 76, No. 7, pp. 5320-5363, 2020, doi: 10.1007/s11227-019-02945-z.

[7] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks", *Engineering Applications of Artificial Intelligence*, Vol. 123, Art. 106432, 2023, doi: 10.1016/j.engappai.2023.106432.

[8] V. Vajrobol, G. J. Saxena, A. Pundir, S. Singh, B. Gupta, A. Gaurav, *et al.*, "Identify spoofing attacks in Internet of Things (IoT) environments using machine learning algorithms", *Journal of High-speed Networks*, Vol. 31, No. 1, pp. 61-70, 2025, doi: 10.3233/JHS-240032.

[9] S. Gvozdenovic, J. K. Becker, J. Mikulskis, and D. Starobinski, "IoT-Scan: Network reconnaissance for Internet of Things", *IEEE Internet of Things Journal*, Vol. 11, No. 8, pp. 13091-13107, 2023, doi: 10.1109/JIOT.2023.3340051.

[10] A. F. Otoom and E. E. Abdallah, "Deep learning for accurate detection of brute force attacks on IoT networks", *Procedia Computer Science*, Vol. 220, pp. 291-298, 2023, doi: 10.1016/j.procs.2023.03.039.

[11] V. P. PM and S. Soumya, "Advancements in anomaly detection techniques in network traffic: The role of artificial intelligence and machine learning", *Journal of Scientific Research and Technology*, Vol. 2, No. 1, pp. 38-48, 2024, doi: 10.5555/jsrt.2024.11.

[12] N. Abid, "Enhanced IoT Network Security with Machine Learning Techniques for Anomaly

Detection and Classification", *International Journal of Current Engineering and Technology*, Vol. 13, No. 6, pp. 536-544, 2023, doi: 10.14741/ijcet/v.13.6.4.

[13] M. Rodriguez, D. P. Tobon, and D. Munera, "A framework for anomaly classification in Industrial Internet of Things systems", *Internet of Things*, Vol. 29, Art. 101446, 2025, doi: 10.1016/j.iot.2024.101446.

[14] Y. Abudurexiti, G. Han, F. Zhang, and L. Liu, "An explainable unsupervised anomaly detection framework for Industrial Internet of Things", *Computers and Security*, Vol. 148, Art. 104130, 2025, doi: 10.1016/j.cose.2024.104130.

[15] I. Katib, E. Albassam, S. A. Sharaf, and M. Ragab, "Safeguarding IoT consumer devices: Deep learning with TinyML driven real-time anomaly detection for predictive maintenance", *Ain Shams Engineering Journal*, Vol. 16, No. 2, Art. 103281, 2025, doi: 10.1016/j.asej.2024.103281.

[16] S. A. Wahab, S. Sultana, N. Tariq, M. Mujahid, J. A. Khan, and A. Mylonas, "A multi-class intrusion detection system for DDoS attacks in IoT networks using deep learning and transformers", *Sensors*, Vol. 25, No. 15, Art. 4845, 2025, doi: 10.3390/s25154845.

[17] A. N. Gummadi, J. C. Napier, and M. Abdallah, "XAI-IoT: An explainable AI framework for enhancing anomaly detection in IoT systems", *IEEE Access*, Vol. 12, pp. 71024-71054, 2024, doi: 10.1109/ACCESS.2024.3400587.

[18] U. Tahir, M. K. Abid, M. Fuzail, and N. Aslam, "Enhancing IoT security through machine learning-driven anomaly detection", *Vfast Transactions on Software Engineering*, Vol. 12, No. 2, pp. 1-13, 2024, doi: 10.21015/vtse.v12i2.1764.

[19] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms", *Sensors*, Vol. 24, No. 2, Art. 713, 2024, doi: 10.3390/s24020713.

[20] R. Rajaan, L. Kumar, N. Choudhary, A. Sharma, and M. Butwall, "Anomaly Detection in Smart Home IoT Systems Using Machine Learning Approaches", *International Journal of Engineering, Business and Management*, Vol. 9, No. 2, Art. 617051, 2025, doi: 10.5555/ijebm.2025.20.

[21] Z. Zulfiqar, S. U. Malik, S. A. Moqurrab, Z. Zulfiqar, U. Yaseen, and G. Srivastava, "DeepDetect: An innovative hybrid deep learning framework for anomaly detection in IoT networks", *Journal of Computational Science*, Vol. 83, Art. 102426, 2024, doi: 10.1016/j.jocs.2024.102426.

[22] M. M. Khan and M. Alkhathami, "Anomaly detection in IoT-based healthcare: Machine learning for enhanced security", *Scientific Reports*, Vol. 14, No. 1, Art. 5872, 2024, doi: 10.1038/s41598-024-56360-1.

[23] A. Albasir, K. Naik, and R. Manzano, "Toward improving the security of IoT and CPS devices: An AI approach", *Digital Threats: Research and Practice*, Vol. 4, No. 2, pp. 1-30, 2023, doi: 10.1145/3579998.

[24] I. Mutambik, "An efficient flow-based anomaly detection system for enhanced security in IoT networks", *Sensors*, Vol. 24, No. 22, Art. 7408, 2024, doi: 10.3390/s24227408.

[25] L. A. Muñoz, J. V. B. Martínez, F. M. Pérez, and I. L. Fonseca, "Anomaly detection system for data quality assurance in IoT infrastructures based on machine learning", *Internet of Things*, Vol. 25, Art. 101095, 2024, doi: 10.1016/j.iot.2023.101095.

[26] W. Shao, Y. Wei, P. Rajapaksha, D. Li, Z. Luo, and N. Crespi, "Low-latency dimensional expansion and anomaly detection empowered secure IoT network", *IEEE Transactions on Network and Service Management*, Vol. 20, No. 3, pp. 3865-3879, 2023, doi: 10.1109/TNSM.2023.3265882.

[27] A. Rahim, Y. Zhong, T. Ahmad, S. Ahmad, P. Plawiak, and M. Hammad, "Enhancing smart home security: Anomaly detection and face recognition in smart home IoT devices using logit-boosted CNN models", *Sensors*, Vol. 23, No. 15, Art. 6979, 2023, doi: 10.3390/s23156979.

[28] Canadian Institute for Cybersecurity, "CIC IoT Dataset 2023", [Online]. Available: https://www.unb.ca/cic/datasets/iotdataset-2023.html, 2023, doi: 10.5555/cic.iot.2023.

[29] H. Kim, B. S. Lee, W. Y. Shin, and S. Lim, "Graph anomaly detection with graph neural networks: Current status and challenges", *IEEE Access*, Vol. 10, pp. 111820-111829, 2022, doi: 10.1109/ACCESS.2022.3215582.

[30] T. Altaf, X. Wang, W. Ni, G. Yu, R. P. Liu, and R. Braun, "GNN-based network traffic analysis for the detection of sequential attacks in IoT", *Electronics*, Vol. 13, No. 12, Art. 2274, 2024, doi: 10.3390/electronics13122274.

[31] S. Ullah, J. Ahmad, M. A. Khan, M. S. Alshehri, W. Boulila, A. Koubaa, *et al.*, "TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT networks", *Computer Networks*, Vol. 237, Art.

110072, 2023, doi: 10.1016/j.comnet.2023.110072.
[32] S. Garcia, A. Parmisano, and M. J. Erquiaga, "IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic", *Zenodo Dataset*, Vol. 1.0.0, pp. 1-23, 2020, doi: 10.5281/zenodo.4743746.