Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.084273

ARTICLE

APENet: Advanced Cyber Security Attack Detection with Attentive Path-Encoding in IoT Networks Using SHAP Based Explainability

Muhammad Mujahid[1], Fatima Alshannaq[1], Shaha Al-Otaibi[2] and Tanzila Saba[1,*]

[1]Artificial Intelligence & Data Analytics Lab, CCIS, Prince Sultan University, Riyadh, Saudi Arabia

[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

*Corresponding Author: Tanzila Saba. Email: tsaba@psu.edu.sa

Received: 19 April 2026; Accepted: 03 June 2026; Published: 13 August 2026

ABSTRACT: Cybersecurity threats in Internet of Things (IoT) networks have escalated, enabled by rapid advancements in wireless communication and edge computing technologies. These advancements expose networks to a wide range of sophisticated and evolving threats and increasingly complex research challenges. Traditional Intrusion Detection and Prevention Systems (IDS/IPS) often fail to provide reliable performance regarding the flexibility and scalability required to handle evolving attack patterns. This study proposes an APENet approach to detect cyberattacks from a real-world cybersecurity dataset, and incorporated a contextual dependency mechanism. The approach captures both local transition dependencies and global relational interactions within structural sequences using bidirectional contextual aggregation and global attention-based interaction modeling. Furthermore, a protocol-aware feature normalization and preprocessing pipeline is developed, including hex-to-integer protocol field normalization, timestamp rebasing, and derivation of traffic dynamics indicators. Severe imbalance in the IoT cybersecurity dataset is addressed with synthetic minority oversampling and TomekLinks techniques to ensure balanced and representative training data. The proposed approach's generalization and robustness are evaluated using stratified 5-fold cross-validation to ensure reliable performance across heterogeneous IoT scenarios. We did several experiments, and the proposed approach achieved remarkable performance for attack detection and underscored the model's flexibility in adapting to various IoT environments. Furthermore, SHapley Additive exPlanations (SHAP) are incorporated to provide explainability, enabling the identification of key features influencing attack predictions and improving trust in model decisions. Overall, this study contributes a robust and adaptive security solution for strengthening IoT ecosystems against evolving cyber threats.

KEYWORDS: Intrusion detection; cybersecurity; explainability; imbalance data; IoT; artificial intelligence

1 Introduction

The rapid growth of IoT devices has significantly made modern network infrastructures more open to cybersecurity risk, and create challenges in resource limited settings [1]. Different communication protocols are used in current era, however Message Queuing Telemetry Transport (MQTT) are lightweight and standard, does not need a lot of bandwidth [2]. This nature also introduce substantial security weaknesses, making MQTT-based IoT systems highly vulnerable to cyberattacks. Denial of service (DOS) and Bruteforce in IoT are critical attacks target the availability of MQTT brokers by overwhelming them with excessive connection requests. Bruteforce attacks attempt to exploit weak authentication mechanisms by trying out every possible combination of credentials in order to gain access to IoT devices/broker services [3,4].

The complexity of data generated by IoT devices requires robust security measures to mitigate the risk of cyberattacks [5]. Traditional security services, which are often passive, struggle to prevent these growing threats and typically only deal with aggregated data [6].

To address these challenges, intrusion detection systems designed for IoT networks that utilize MQTT must consider the dynamic, heterogeneous, and temporal nature of network traffic [7]. One significant challenge in existing studies is the heavy reliance on simulated datasets. These types of IoT datasets often fail to capture the complexity of real-world device behavior and attack patterns. The collected MQTTEEB-D dataset offers a more realistic and representative benchmark, as it is derived from a real IoT deployment. Cyberattack detection systems mostly relied on traditional machine learning (ML) and deep learning (DL) models to tackle these challenges. Traditional approaches are known to not perform well with data with many variables and limit generalization. The traditional model also struggles in dealing with unknown attack variants when the dataset is imbalanced. In these days, DL models have proved better with large datasets and have the potential to improve IDS capabilities significantly. The study makes the following key contributions:

- This work presented a unified feature structural framework for IoT based MQTT traffic, encodes packet-level statistical attributes, and decision semantics, enabling integrated analysis of network flow behavior across heterogeneous feature spaces. The transformation technique is incorporated to map hierarchical decision traces into fixed-length sequential index representations.
- Introduced controlled stochastic masking mechanism applied to structural traces, simulating partial observation conditions and improving robustness against noisy flow-level observations.
- The work used protocol-aware feature normalization, preprocessing pipeline is developed, including hex-to-integer protocol field normalization, timestamp re-basing, and derivation of traffic dynamics indicators such as normalized transmission rate and tail-heavy packet identification. Contextual dependency mechanism that captures both local transition dependencies and global relational interactions within structural sequences using bidirectional contextual aggregation and global attention-based interaction modeling.
- The experimental MQTTEEB-D real time security dataset is extreme imbalance in attack classes, synthetic data generation mechanism is employed to address imbalance, improving learning from rare attack patterns and data diversity. Also, 5Fold cross validation K = 5 is incorporated to further validate the model performance.

The remaining paper follows the literature review in Section 2, Proposed methodology in Section 3, results and discussion are presented in Section 4, and in last Section 5 describes the conclusion and future work.

2 Literature Review

Intrusion and anomaly detection in MQTT based IoT networks has now moved to ML and DL frameworks, instead of conventional approaches. The work integrated elliptic curve cryptography with artificial neural networks (ANN) and used different ML classifiers. Their approach maintained confidentiality with elliptic curve cryptography, and ANN attained 90.3% accuracy. Their approach is trained on limited attack types and may fail to see unseen attacks [8]. The stacked model was presented by [9] integrated with ID model, processed and trained on an MQTT attacks dataset. They used different ML models and averaged the outputs to make final results. The author incorporated bagging, boosting, and stacking. Their findings show a 95% F1 score. Ullah et al. [10] proposed an intrusion detection framework using a transformer neural network, with the ultimate goal of improving the detection of malicious activity in MQTT-enabled IoT networks. Experimental results show that the proposed DS could significantly improve the detection

of malicious activity in such networks. The performance of the proposed method was evaluated on the MQTT-IoT-IDS2020 dataset.

An AI framework to secure IoT systems was presented by [11], used predictive analytics and prevention techniques. They demonstrated the importance of realistic traffic for training robust intrusion detection models. The pre-trained models performed poorly, while retraining DT and GRU on MQTTEEB-D yielded consistent improvements. GRU attained 86% and DT attained 87% accuracy. The main drawback is applying SMOTE on the entire dataset, which may cause data leakage for the validation process, and it does not employ any explainability of either ML or DL models for gaining trust. Similarly, Allaga et al. [12] incorporated Ml framework for anomaly detection on MQTTEEB-D, for original and balanced datasets, used various metrics including MCC and attained highest 98.8% accuracy. They claim that simple models based on tree are more robust while complex models are not robust for imbalance datasets. Subsequently, the author proposed integrated approach for threat detection using two real time IoT datasets. On MQTTEEB-D, their voting ensemble approach attained 92.68% accuracy. They did not used learning curves, ROC-AUC, and other metrics to validate the results [13].

Recently, the IoT networks have grown exponentially, and securing these networks is challenging. So, a robust intelligent intrusion detection framework was presented by [14] for the prediction of cyber-threats through elite ML classifiers. Employed lightweight models, but feature generation could take time and considered complex for IoT environment. The dataset consisted on normal, attack on a subscriber, and attack on a broker. Experiment using different ML models and attained competitive results. Santos et al. [15] presented an intrusion detection method to protect IoT networks from external and internal threats. The proposed framework monitored and detected attacks, intrusions, and other types of anomalies at different IoT architectures based on some flow features. Schrötter et al. [16] used a DNN model for intrusion detection and attained the best performance, but when applied to another similar type of dataset, its performance decreased. For binary classification, the models performed better than others. Zeghida et al. [17] proposed a unique problem solution to balance the datasets with Generative Adversarial Networks that generate synthetic samples and increase the performance. Also, utilized hybrid DL models for attack detection on generated balanced data. The authors proposed an intrusion detection system that used a genetic technique for the selection of important features and then employed a neural network for the training and classification of the network packets [18]. The MQTT attacks-based public dataset was used by [19] for the detection of attacks with DL models, standard metrics were used for the performance, and the best performance attained was 97%. The author proposed DNN for intrusion detection in the MQTT-based protocol and also compared its performance with other traditional ML models [20]. Key strengths and limitations regarding the previous work done by the authors for intrusion and attacks detection are presented in Table 1.

Table 1: Comparative literature review analysis for intrusion detection.

Key Technique Used	Key Strengths	Key Limitations
ANN, DT, NB [8]	Cybersecurity attack detection, MQTT based communication, provide lightweight and resource constrained fraemwork	Low accuracy for six attack types with simple ANN model, did not provide model explainability, and limited dataset
Ensemble, Stacking, AdaBoost [9]	To secure MQTT protocol with high performance, ensemble of several classifiers for better accuracy and generalization	Performed and visualized binary classification results, did not employed explainability via SHAP or other techniques

(Continued)

Table 1 (continued)

Key Technique Used	Key Strengths	Key Limitations
TNN-IDS, DBN [10]	Very high prediction accuracy, captured complex patterns, improved minority class	High computational cost of transformers, difficult to deploy, no confusion matrix for error analysis
LightGBM, DT, GRU [11]	Retrain the models, best performance, secure IoT system, balance the datasets, consistent improvements	Poor generalization, SMOTE may cause data leakage, interpretability not performed
Comprehensive ML [12]	High accuracy, reduction in training time, balanced data, performance efficacy, tree based simple model	Weak ML models, ignored, advance DL model, No explainability analysis
Voting ensemble, bagging, RF [13]	Presents overall resource consumption score, handle missing data, improve generalization	Missing learning curves, ROC-AUC, interpretability, cross validation, low accuracy
SVM, DT, RF, GB [14]	Data collection, feature generation, quality parameters, high accuracy, lightweight models	No explainability, perfect accuracy may cause overfitting, weak generalization, high selection time

3 Methodology

This study used an advanced and unique pipeline for the classification of real-time cyberattacks on 222, 813 samples in MQTT-IoT networks. The study presented APENet, a learning approach that integrates tree structural knowledge with deep temporal and attention-based modeling. The approach has path regularization through stochastic dropout, embeddings, a positional encoding mechanism, temporal modeling, an attention module, the fusion of features, and the classification of attacks. All the samples are preprocessed, normalized and then distributed into train and test. The imbalanced cyberattacks in MQTT networks are balanced only for train data via SMOTE and TomekLink. After that, the APENet approach is employed for classification; the entire pipeline is shown in Fig. 1 and evaluated using various performance metrics.

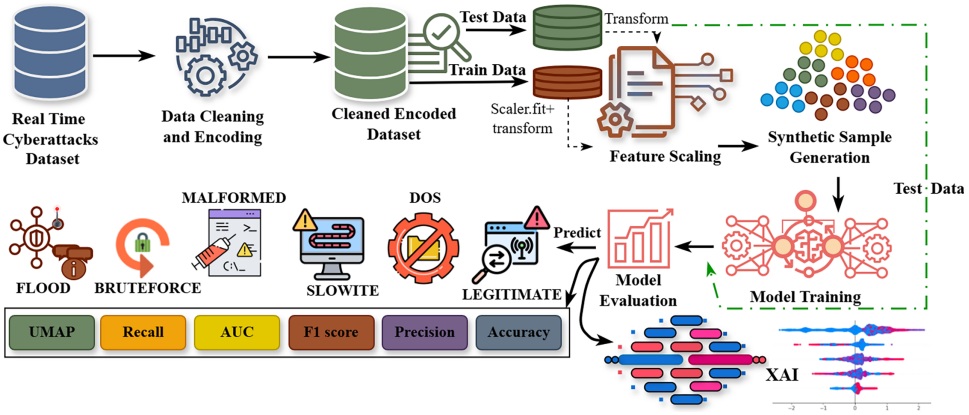

Figure 1: Comprehensive workflow of the proposed approach for targeting cyberattack classification with interpretability.

3.1 Dataset Description

Real-world IoT cybersecurity dataset [21] MQTTEEB-D consists 222,813 raw entries and 13 features including target label. Raspberry Pi, IoT health sensors, and MQTT broker server used to represent the dataset. Dataset can be defined as (Eq. (1)):

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}, \quad x_i \in \mathbb{R}^d, \; y_i \in \{1, \ldots, C\} \tag{1}$$

x_i indicate the input feature vector and y_i represents the class label (target) to make predictions.

The dataset includes DoS (41,802), Slowite (20,090), Malformed (105,528), BruteForce (27,377), flooding (10,213) and legitimate (17,557) attacks. There is extreme class imbalance in MQTTEEB-D dataset, train data after split is represented in Fig. 2a. The train data after TomekLinks technique is shown in Fig. 2b. The train data after SMOTE technique is shown in Fig. 2c. The train data after SMOTE + TomekLinks technique is shown in Fig. 2d, and this approach balances oversampling and boundary cleaning.

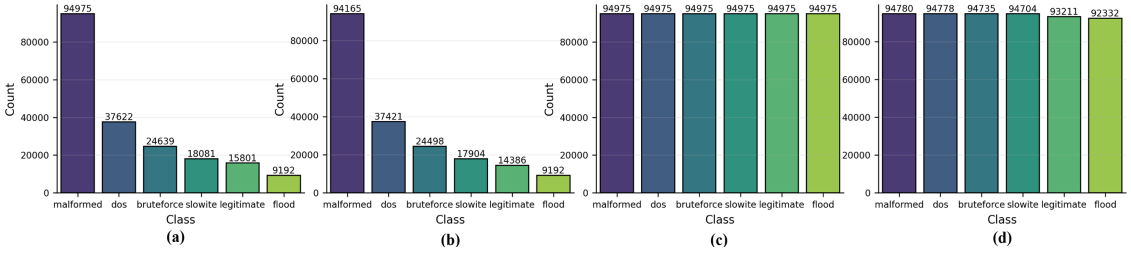

Figure 2: Comprehensive details of the MQTTEEB-D dataset: (**a**) distribution of cyberattacks for train data, (**b**) distribution of TomekLinks train data (under-sampling), (**c**) distribution of SMOTE train data (over-sampling), (**d**) distribution of SMOTE + TomekLinks train data.

3.2 Data Preprocessing and Normalization

To remove meaningless information from the dataset for better model predictions, preprocessing is crucial as the model cannot perform with raw data. The dataset contains NaN values that are needed to remove or fill with some techniques. The features are converted into numeric representation. Invalid label are removed as in Eq. (2):

$$y_i = \begin{cases} \text{valid}, & \text{if } y_i \notin \{\text{nan, none, unknown}\} \\ \text{removed}, & \text{otherwise} \end{cases} \tag{2}$$

The dataset has different feature such as tcp flags, tcp length, mqtt msg, etc and we convert it into numeric form; also missing or infinite values are handled as in Eq. (3). Next, feature standardization is incorporated, where μ and σ represents mean and standard deviation of each feature, also in Eq. (4).

$$x_i^{(j)} = \begin{cases} 0, & \text{if } x_i^{(j)} \in \{\text{NaN}, \pm\infty\} \\ x_i^{(j)}, & \text{otherwise} \end{cases} \tag{3}$$

$$\tilde{x}_i = \frac{x_i - \mu}{\sigma} \tag{4}$$

In some scenarios, the categorical and MQTT/TCP fields were safely converted to numeric representations; temporal and traffic features were derived from timestamps. SMOTE is used in this work to handle

class imbalance, and TomekLinks in MQTTEEB-D training data only. This synthetically generated samples for minority class, select nearest neighbors and generate new data as described in Eq. (5).

$$x_{\text{new}} = x_i + \lambda(x_{nn} - x_i), \quad \lambda \sim U(0,1) \tag{5}$$

3.3 Tree Based Structural Encoding

Tree based supervised learning approach is used for classification for predicting the target class (different attacks). For trees, little data preprocessing is required. An individual tree $\mathcal{T}$ is trained (Eq. (6)). This used full paths for each sample instead relying only on leaf nodes. For inputs path, n_k represents the k-th visited node and L_i is the path length.

$$\begin{aligned} &\mathcal{T} : \mathbb{R}^d \to \mathcal{Y}, \\ &P_i = \{n_1, n_2, \ldots, n_{L_i}\} \end{aligned} \tag{6}$$

The sequences are padded for the batch processing, as each path from root to leaf is represented as an ordered sequence of nodes. Zero-padding is used to standardize paths to a constant length because they vary in length.

$$\hat{P}_i = \{n_1, n_2, \ldots, n_{L_i}, 0, \ldots, 0\}, \quad |\hat{P}_i| = L \tag{7}$$

3.4 Path Regularization via Stochastic Dropout

Path regularization using dropout is also incorporated into sequences of tree nodes and prevents overfitting. After fitting, the input sample is mapped to its traversal path in the tree. Then paths are padded to a fixed length suitable for the next processing. Generates a random mask like in Eq. (8) of the same shape as the sequence array; each element has drop_prob. Here, p denotes the dropout probability.

$$\tilde{P}_i^{(j)} = \begin{cases} 0, & \text{if } r < p \\ \hat{P}_i^{(j)}, & \text{otherwise} \end{cases} \quad r \sim U(0,1) \tag{8}$$

3.5 Path Embedding and Positional Encoding

After stochastic dropout, padded paths are converted into an embedding layer, mapping node index in the sequence to a dense vector. At this stage, the network learns similarities between nodes that have the same decision roles. Next, positional encoding are fed into the embeddings to preserve node order. To encode positional information, sinusoidal positional encoding is added.

$$\begin{aligned} &e_j = \mathbf{E}(n_j), \quad \mathbf{E} \in \mathbb{R}^{|\mathcal{N}| \times d_e} \\ &PE(pos, 2k) = \sin\left(\frac{pos}{10000^{2k/d_e}}\right) \\ &PE(pos, 2k+1) = \cos\left(\frac{pos}{10000^{2k/d_e}}\right) \\ &z_j = e_j + PE(j) \end{aligned} \tag{9}$$

3.6 Temporal Modeling and Attention Mechanism

The generated configuration is passed through the bidirectional unit that maintains the vertices along the path. By processing the sequence of points in both forward and backward directions, unit allows the classification of each node to incorporate information from previous and subsequent nodes, while layer

processing improves training by preserving as many of its features as possible. Next, multiheader simulation is used, which helps the model focus on the most important components and learn global relationships along the path. Residual connectivity, along with feature regularization, improves feature decomposition and helps the network learn a more robust representation.

After attention, GAP reduces the cross-sectional variance to a vector that is constant along the entire tree path. This pathway representation is then mapped onto the original features. For learning high-level feature interactions and improving the chances of reducing overfitting, dense layers with ReLU and dropout are utilized. At the end, softmax with six numbers of classes produces predictions. High-level APENet architecture with key modules for cyberattack classification is represenetd in Fig. 3.

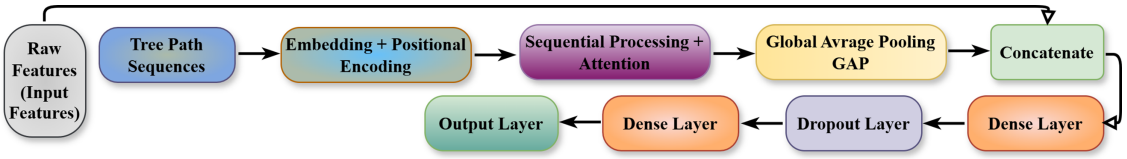

Figure 3: High-level APENet architecture with key modules for cyberattack classification.

Overall, in short, for this approach, converts the tree paths into sequences, node embeddings, and then applied stochastic path dropout that is better for regularization and prevents from overfitting. After this, include positional encoding, incorporated bidirectional gated recurrent unit, attention, layer normalization, pool sequences and then concatenate with original features. Next, applied dense, dropout and then final sofmax layer for predictions. The model is trained using categorical crossentropy loss with 0.1 label smoothing, Adam for optimizer (learning rate = 1×10^{-3}), and key performance metrics. The work employed total 15 epochs, 256 batch size, and tensorflow python. The proposed methodology is shown in Algorithm 1.

$$\mathcal{L}\& = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log(\hat{y}_{ic}) \tag{10}$$

Eq. (10) represents categorical cross-entropy loss used for classification. It measures the difference between true labels y_{ic} and predicted probabilities $\hat{y}_{ic}$ across all samples and classes. The loss penalizes incorrect predictions by taking the negative log of the predicted probability of the true class.

Algorithm 1: APENet: Proposed attentive path encoding network for cyberattacks

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, $x_i \in \mathbb{R}^d$, $y_i \in \{1, \ldots, C\}$
Ensure: Trained APENet, predictions $\hat{y}_i$
1: **Cleaning:** Remove missing, NaN, or invalid entries
2: **Numeric Conversion:** Convert features to numeric, replace NaN with 0 or median:
$X \leftarrow \text{nan\_to\_num}(X)$
3: **Label Encoding:** $y \leftarrow \text{LabelEncoder}(y)$
4: Convert y to one-hot: $y_{\text{cat}} = \text{to\_categorical}(y)$
5: **Train/Test Split:** Split $\tilde{X}$, y_{cat} into training and test sets
6: **Oversampling:** Apply sampling on training set
7: **Train Tree model:** $T \leftarrow \text{Classifier}()$, $T.fit(X_{\text{train}}, y_{\text{train}})$
8: **Extract decision paths:**

(Continued)

Algorithm 1 (continued)

for each sample x_i in training/test sets **do**
$p_i \leftarrow$ indices of nodes visited in $T(x_i)$
end for
Pad paths: Ensure all paths have length $L = \max_i |p_i|$
Apply path dropout: Randomly mask nodes with probability p_d
Tree embedding: $e_i = \text{Embedding}(p_i) \in \mathbb{R}^{L \times E}$
Positional encoding:
$PE_{j,k} = \begin{cases} \sin\left(\frac{j}{10000^{2k/E}}\right) & k \text{ even} \\ \cos\left(\frac{j}{10000^{2k/E}}\right) & k \text{ odd} \end{cases}$
$e_i \leftarrow e_i + PE$
Bidirectional gradient: $h_i = \text{gradient}(e_i) \in \mathbb{R}^{L \times 2H}$
Layer Normalization: $h_i \leftarrow \text{LayerNorm}(h_i)$
Multi-Head Self-Attention:
$A = \text{MultiHeadAttention}(h_i, h_i)$
$h_i \leftarrow \text{LayerNorm}(h_i + A)$
Global Pooling: $g_i = \text{GlobalAveragePooling1D}(h_i) \in \mathbb{R}^{2H}$
Feature Fusion: Concatenate g_i with raw features: $f_i = [g_i, x_i]$
$f_i \leftarrow \text{ReLU}(\text{Dense}_{128}(f_i))$
$f_i \leftarrow \text{Dropout}(f_i, 0.2)$
$f_i \leftarrow \text{ReLU}(\text{Dense}_{64}(f_i))$
$\hat{y}_i = \text{Softmax}(\text{Dense}_C(f_i))$
Compile Model: Cross-entropy, label smoothing, optimizer = Adam
Train Model: Fit on $[X_{\text{train}}, p_{\text{train}}]$, validate on $[X_{\text{test}}, p_{\text{test}}]$
Evaluate: Test set

4 Results and Discussion

This section represents the comprehensive evaluation of the APENet approach, ML approaches, synthetic data generation, performance comparison with traditional and state of the art models, interpretability of the proposed approach and visualizations. The experiments are conducted on real world MQTTEEB-D dataset.

4.1 Experiments under Stratified 5Fold Cross Validation for Balanced MQTTEEB-D

Table 2 represents the comprehensive evaluation of traditional models with the proposed APENet model under 5Fold cross validation and synthetic generated dataset (SMOTE) in first experiment. We ensured that each fold preserves temporal ordering, meaning that training data always precedes test data chronologically. The eXtreme gradient boosting model attained highest recall, while LDA-Base lowest F1 score. Similarly, other models performance reached to 90%. But the APENet model attained superior performance on (SMOTE) data. The evaluation demonstrate that best performance on balanced data attained by the proposed model.

Table 2: Comprehensive analysis of traditional linear, boosting, neural models and proposed APENet approach via balanced MQTTEEB-D under 5Fold cross validation.

Models	ACCS	PRECS	RECS	F1S	AUCS	MCCS
XGBM-BalCV	91.51 ± 0.0025	91.20 ± 0.0026	91.51 ± 0.0025	91.33 ± 0.0026	98.27 ± 0.0005	88.00 ± 0.0035
CaTBM-BalCV	91.36 ± 0.0011	91.02 ± 0.0004	91.36 ± 0.0011	91.15 ± 0.0007	98.24 ± 0.0003	87.80 ± 0.0013
LGBM-BalCV	90.79 ± 0.0028	91.19 ± 0.0020	90.79 ± 0.0028	90.96 ± 0.0024	98.25 ± 0.0004	87.09 ± 0.0035
MLPN-BalCV	75.72 ± 0.0085	78.04 ± 0.0046	75.72 ± 0.0085	76.04 ± 0.0064	94.91 ± 0.0016	67.16 ± 0.0081
LDAM-BalCV	51.85 ± 0.0023	53.59 ± 0.0010	51.85 ± 0.0023	51.28 ± 0.0015	76.64 ± 0.0020	31.33 ± 0.0027
LLRM-BalCV	54.77 ± 0.0022	58.09 ± 0.0025	54.77 ± 0.0022	54.63 ± 0.0022	80.32 ± 0.0008	40.52 ± 0.0027
Proposed	94.95 ± 0.0031	95.27 ± 0.0027	94.95 ± 0.0031	95.07 ± 0.0029	97.88 ± 0.0057	92.92 ± 0.0043

4.2 Experiments 5Fold Cross Validation and Hold-Out Settings for Proposed APENet

In second experiments, the MQTTEEB-D dataset is distributed into five equal folds using cross validation, with each fold has same class proportions. This is required to ensure reliable and unbiased performance, Table 3 presents the results under cross validation with synthetic data with TomekLinks. The results demonstrate that approach attained 98.65% accuracy highest for fold 4, and 99.55 AUC, 98.58 ± 0.0708 mean accuracy and standard deviation (STD).

Table 3: Performance analysis with cross validation 5Folds for proposed APENet approach.

CV-Folds	ACCS	PRECS	RECS	F1 Score	AUCS	MCCS
CVFO1	98.46	98.46	98.46	98.46	99.38	97.83
CVFO2	98.63	98.62	98.63	98.63	99.35	98.07
CVFO3	98.62	98.61	98.62	98.61	99.25	98.05
CVFO4	98.65	98.64	98.65	98.65	99.55	98.10
CVFO5	98.54	98.53	98.54	98.53	99.43	97.94
Mean + STD	98.58 ± 0.0708	98.57 ± 0.0700	98.58 ± 0.0708	98.58 ± 0.0702	99.39 ± 0.1000	97.99 ± 0.1000

In third experiment, Table 4 presents the attack wise performance analysis of proposed APENet approach via TomekLinks MQTTEEB-D with Hold-Out setting. The attack wise performance with two different splits, one test size 0.1 and second test size 0.2 are utilized. BRUTEFORCE class attained remarkable precision for test size 0.1 and 0.2, 99.53% and 99.34%, respectively. LEGITIMATE class attained poor performance as compared to other attacks. The APENet approach attained 98.85% weighted average precision with 0.1 test set and 98.74% for 0.2 test set. Overall, the proposed approach is well structured and attained superior performance using Hold-Out setting with TomekLinks data.

Table 4: Attack wise performance analysis of proposed APENet approach via TomekLinks MQTTEEB-D with hold-out setting.

Attacks	PRECS (Test 0.1)	RECS (Test 0.1)	F1S (Test 0.1)	PRECS (Test 0.2)	RECS (Test 0.2)	F1S (Test 0.2)
BRUTEFORCE	99.53	99.49	99.51	99.34	99.31	99.32
DOS	99.21	99.52	99.37	99.19	99.34	99.27
FLOOD	98.24	98.24	98.24	98.01	98.83	98.42
LEGITIMATE	93.46	92.77	93.11	93.91	91.37	92.62
MALFORMED	99.45	99.58	99.52	99.34	99.57	99.46
SLOWITE	99.00	98.36	98.68	98.49	98.90	98.70

(Continued)

Table 4 (continued)

Attacks	PRECS (Test 0.1)	RECS (Test 0.1)	F1S (Test 0.1)	PRECS (Test 0.2)	RECS (Test 0.2)	F1S (Test 0.2)
Macro avg	98.15	97.99	98.07	98.05	97.89	97.96
Weighted avg	98.85	98.85	98.85	98.74	98.76	98.75

Fig. 4 illustrates the comprehensive performance of APENet approach using Hold-Out splitting method with 0.1 test size and TomekLinks data for only train set to improve generalization and prevents from any data leakage. Fig. 4a represents the precision-recall curves for cyberattacks; indicates 98.15 AP for DOS, 98.66 AP for MALFORMED, 98.81 AP for BRUTEFORCE, and 94.97 AP for FLOOD attack. Also, Fig. 4b represents ROC-AUC where DOS (AUC = 99.78), BRUTEFORCE (AUC = 99.78) and LEGITIMATE (AUC = 95.19). The proposed model loss curves are illustrated in Fig. 4c, green curve shows training and orange shows the validation loss. Fig. 4d visualized all cyberattacks via UMAP, it extracts softmax probabilities as feature representations for test samples.

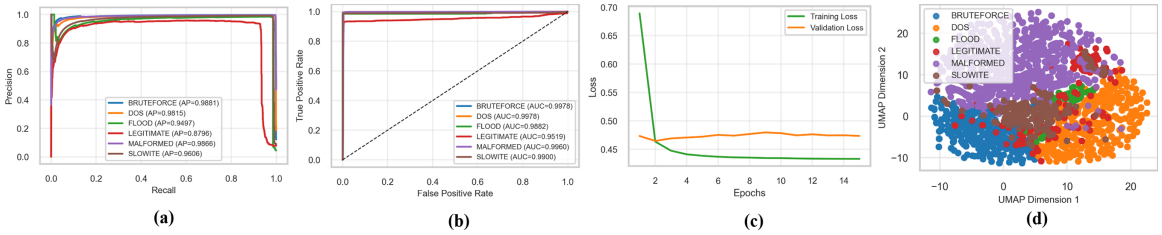

Figure 4: Overall representation performance of APENet approach with TomekLinks data; (**a**) precision-recall curves, (**b**) ROC-AUC curves, (**c**) analysis of loss, (**d**) UMAP for six cyberattacks for last layer (softmax).

4.3 Mis-Classification Analysis

Fig. 5 presents the comprehensive mis-classification analysis, visualize most frequent errors made by APENet approach. It is essential in cyberattack prediction, as it evaluates accuracy and analysis among the malicious and normal attacks. This work compared two classes by their classification errors made by the model.

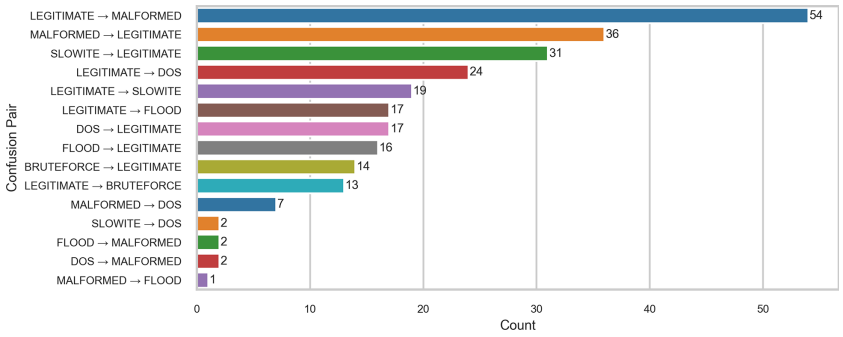

Figure 5: Visualize critical mis-classification pair pattern for each attack class in the proposed APENet approach.

Furthermore, full attack wise error analysis visualized in Fig. 6 using a 2D UMAP projection. Accurate and inaccurate samples are identifies, data is reduced with PCA, and then plotted the six classes. The correct samples as circles and misclassified ones as prominent “X” markers.

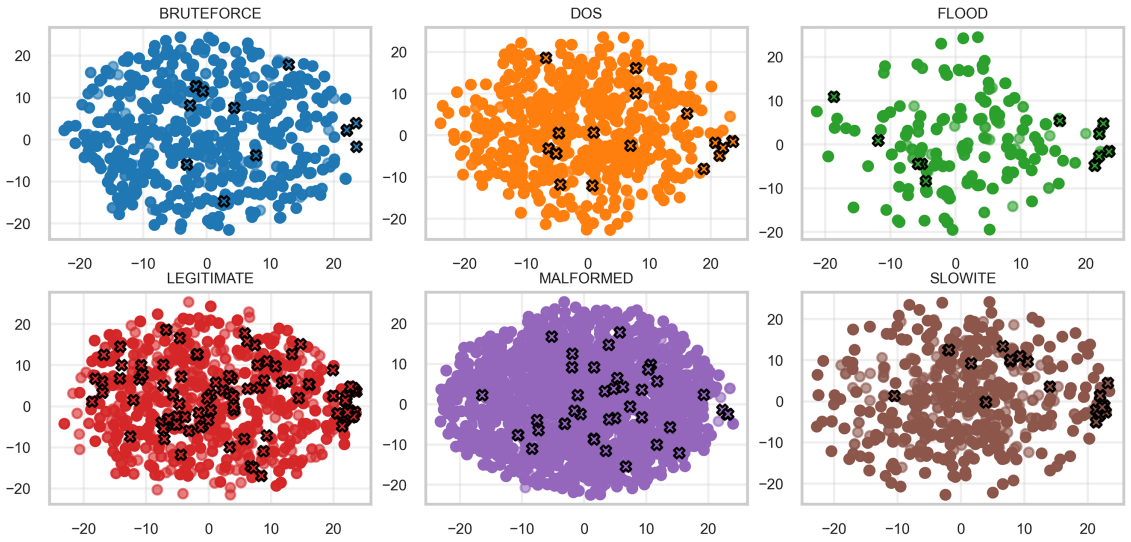

Figure 6: Attack wise UMAP visualization with highlighted error analysis using proposed APENet approach.

4.4 Analysis SHapley Additive exPlanations

This work uses SHAP, an interpretability technique in AI, to explain how each attribute contributes to a particular prediction since it offers precise, grounded explanations. Individual predictions are thoroughly explained by SHAP, while aggregate SHAP values across data show trends in feature importance, comprehending overall model behavior and offering visual interpretability. Fig. 7 represents the SHAP summary plot for each attack class from the MQTTEEB-D dataset. We plot only top five features that contribute most to the model predictions and their relevance to MQTT protocol behavior. The tcp_len highlights packe, size anomalies, mqtt_msg captures abnormal message flow, timestamp tracks temporal attack patterns,, tcp_time_delta irregular inter-packet timing, and mqtt_conflag cleansess. The model's decisions are both meaningful and consistent with real-world MQTT IoT network behavior for cybersecurity applications.

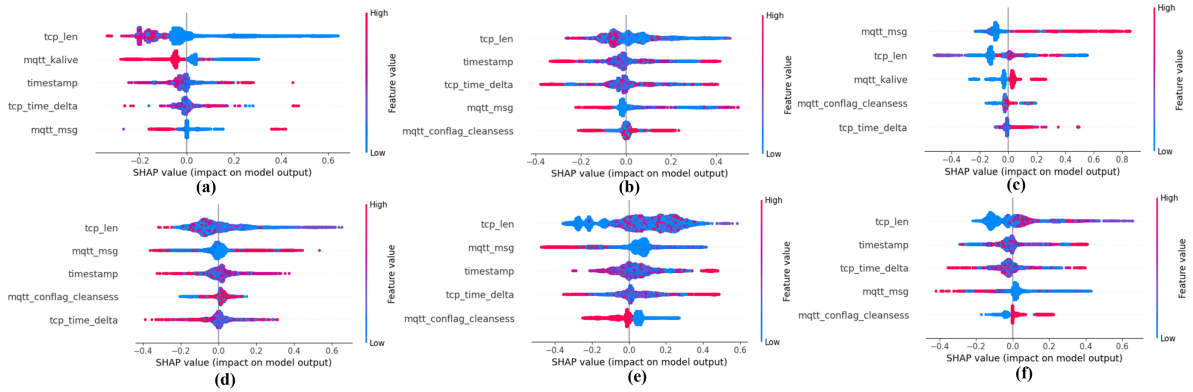

Figure 7: Interpretability using SHAP from the model to illustrate top contributions; (**a**) BRUTEFORCE attack class, (**b**) DOS attack class, (**c**) FLOOD attack class, (**d**) LIGITIMATE class, (**e**) MALFORMED attack class, and (**f**) SLOWITE class. Visualize top 5 features analysis for each class that contribute maximum for the predictions.

In addition, we conducted leave-one-class-out experiment on MQTTEEBD dataset for evaluation of proposed approach efficacy for zero-day attacks as illustrated in Table 5. This experiment used entropy of the predicted probabilities as an uncertainty measure, with high-entropy samples classified as zero-day.

We removed one class from the dataset and retrained the proposed approach on the remaining attacks. While testing the approach, samples from the excluded class were treated as zero-day attacks. We incorporated detection rate (DR), false alarm rate (FAR), and AUC. The experiment achieved minimum FAR.

Table 5: Performance against zero-day attacks.

Attack	DR	FAR	AUC
Bruteforce	0.6738	0.0202	0.9420
DOS	0.9959	0.0144	0.9984
Flood	0.9775	0.0100	0.9890
Malformed	0.5779	0.0070	0.9905
Slowite	0.9268	0.0105	0.9915

4.5 Statistical Significance

The paired test is a statistical method used to compare the performance of different models as shown in Table 6. It is useful when a model's performance has been run multiple times. It examines the difference between each pair of observations. The null hypothesis states that the mean difference is zero, meaning there is no effect or change. A small p-value usually less than 0.05 indicates a statistically significant difference, indicating that one model performs better than another under similar conditions.

Table 6: Statistical paired test.

Comparison	p-Value	Significant
Proposed vs. XGBM	<0.05	yes: ✓
Proposed vs. CaTBM	<0.05	yes: ✓
Proposed vs. LGBM	<0.05	yes: ✓
Proposed vs. MLPN	<0.05	yes: ✓

4.6 Ablation Study

The contribution of each component is evaluated with several experiments. Table 7 represents the results of ablation study. The approach achieved 90.21% accuracy with only raw features, indicating that raw features were insufficient for optimal performance for deep learning. The approach achieved 98.57% accuracy with path embeddings and significantly improved the performance. Path embeddings demonstrate the efficacy of structural feature representations. With an attention mechanism, only a minor improvement was observed. The full APENet model achieved the best performance of 98.85%, validating the effectiveness of raw features, path embeddings, sequential learning, and attention mechanisms.

Table 7: Results of ablation study.

Configuration	Accuracy	Details
Only Raw Features	90.21	Remove path embeddings or attention, include raw features,
Only Path Embeddings	98.57	Remove raw features, include path embeddings or attention
W/O Attention	98.49	Remove attention, include raw features, path embeddings, sequence modeling
Full APENet	98.85	Complete model using raw features + path embeddings, sequence modeling, and attention

4.7 Comparison with State of the Art Methods

The proposed approach is compared with the existing work, using the same dataset to evaluate the efficacy and provide robust generalization. Different authors utilized individual ML or DL models, and some utilized stacking, and voting ensemble for attack detection with MQTT datasets. The comparison with existing work is shown in Table 8. Karimunda et al. [8] employed MQTTSET dataset, but failed to attained best performance. Also had poor performance, and did not provide explainability SHAP analysis. Like, in Zeghida et al. [9] study, stacked model was proposed to attained the better results but also lack of intrepretability. The authors in study [11] also employed MQTTEEB-D dataset with DT model, attained very low 87% accuracy. Similarly, Qasem et al. [13] conducted the research in 2026 by utilizing the ensemble technique and attained 92.6% accuracy. The existing work attained poor results, generalization problems, did not used explainable XAI techniques to gain trust and enhance model predictions. The proposed IDS leverages optimized strong techniques on extracted traffic features to achieve superior detection performance. It improves classification accuracy, reduces false positives, and enhances generalization over traditional techniques and signature-based methods.

Table 8: Comparison of proposed approach with the existing work.

Existing Work	Employed Data	Methods	Accuracy	Remarks
Karimunda et al. [8] [2025]	MQTTSET	ANN	90.3	Poor accuracy, utilized simple ANN, no interpretability
Zeghida et al. [9] [2023]	MQTT Data	Stacking	95.4	Computationally costly, better results, no interpretability
Karam et al. [11] [2025]	MQTTEEB-D	DT	87.0	Very poor performance, no interpretability
Allaga et al. [12] [2025]	MQTTEEB-D	Benchmark ML	98	Good performance, no interpretability
Qasem et al. [13] [2026]	MQTTEEB-D	Voting Ensemble	92.6	Poor performance, no cross validation, no interpretability
Proposed approach [2026]	**MQTTEEB-D**	**APENet**	**98.8**	Good results on balanced data, provide model interpretability, cross validation

4.8 Discussion

The experimental results demonstrate that the proposed approach can effectively identify malicious activities within MQTT-based IoT environments. The highest performance indicates the capability of the model to distinguish between attacks. The model achieved 98.25% with SMOTE, with TomekLinks achieved 98.85%, and SMOTE + TomekLinks achieved 98.06% accuracy. So, results illustrate that TomekLinks effectively removes noisy samples, a slight improvement in performance compared to SMOTE. The SMOTE + TomekLinks balances minority class coverage and boundary cleanliness.

With 5Fold cross-validation, the model achieved 98.5 ± 0.0702 f1 score and 97.99 ± 0.1000 MCC score. The study also conducted a leave-one-class-out experiment on the MQTTEEBD dataset and utilized the entropy of the predicted probabilities as an uncertainty measure. The approach achieved a 99.59% detection rate for DOS attacks and 97.75% for Flood, while Bruteforce and Malformed achieved some poor performance. Another dataset was used for the experiment, the proposed approach achieved 99.04%, 96.54%, 97.68% macro average precision, recall, and f1 score, respectively.

Regarding the suitability for constrained IoT environments, the proposed approach with TomekLinks sampling has 662,278 trainable parameters, a 7.77 MB model size, 5.5132 s total inference time, 0.2477 ms/sample average latency, and an approx. 1.29 MB memory footprint. These suggest that the proposed approach can be efficiently implemented on resource-limited devices and is suited for IoT applications.

5 Conclusion and Future Work

In this work, we presented a novel cyberattack detection approach to secure an MQTT-based IoT environment. The proposed approach integrated preprocessing pipeline and SMOTE and Tomek Links oversampling and undersampling strategies to tackle extreme imbalance data effectively. Cross-validation 5-fold was also applied to ensure robust performance evaluation and minimize overfitting across heterogeneous IoT environments. Furthermore, incorporated SHapley Additive exPlanations enhance model interpretability and highlight the influential features that contribute most to model decisions. This will improve trust and transparency. The approach achieved 94.95 ± 0.31 mean accuracy with STD in second experiment using SMOTE, 98.85% accuracy with 0.1 split, 98.75% accuracy and 98.76% recall with 0.2 split, 98.58 ± 0.07 mean accuracy with STD in third experiment using TomekLinks. Also, the approach attained 99.39 ± 0.10 AUC, and 97.99% MCC score. The findings illustrate that the proposed approach attained high performance with efficient resources and suitable for deployment in constrained IoT environments. Although the proposed framework achieved the best performance utilizing SMOTE and TomekLinks and the APENet model, in the future it could explore transformer-based encoders. The long-range dependencies in sequential IoT data are captured with the self-attention mechanism of transformers and enhance the temporal patterns that the APNet model may not have captured fully. Also, transformers are very scalable due to the parallel computation and dynamic sequence handling.

Acknowledgement: This research is supported by Princess Nourah bint Abdulrahman University, Researchers Supporting Project number (PNURSP2026R136), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors are also thankful to Prince Sultan University, Riyadh, Saudi Arabia of APC support.

Funding Statement: This research is funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R136), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: Conceptualization, Tanzila Saba and Muhamamd Mujahid; methodology, Tanzila Saba; software; validation, Shaha Al-Otaibi and Fatima Alshannaq; formal analysis and investigation, Shaha Al-Otaibi and Fatima Alshannaq; resources, Tanzila Saba; data curation, Fatima Alshannaq; writing—original draft preparation,

Muhammad Mujahid and Tanzila Saba; writing—review and editing, Tanzila Saba, Fatima Alshannaq, and Shaha Al-Otaibi; visualization, supervision, Tanzila Saba; project administration, Tanzila Saba; funding acquisition, Tanzila Saba. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Reyes-Acosta RE, Mendoza-González R, Oswaldo Diaz E, Vargas Martin M, Luna Rosas FJ, Martínez Romo JC, et al. Cybersecurity conceptual framework applied to edge computing and internet of things environments. Electronics. 2025;14(11):2109. doi:10.3390/electronics14112109.
2. Yassein MB, Shatnawi MQ, Aljwarneh S, Al-Hatmi R. Internet of things: survey and open issues of MQTT protocol. In: Proceedings of the 2017 International Conference on Engineering & MIS (ICEMIS); 2017 May 8–10; Monastir, Tunisia. p. 1–6.
3. Laghari SUA, Li W, Manickam S, Nanda P, Al-Ani AK, Karuppayah S. Securing MQTT ecosystem: exploring vulnerabilities, mitigations, and future trajectories. IEEE Access. 2024;12(7):139273–89. doi:10.1109/ACCESS.2024.3412030.
4. Hashimyar ME, Aiash M, Khoshkholghi A, Nalli G. Signature-based security analysis and detection of IoT threats in advanced message queuing protocol. Network. 2025;5(1):5. doi:10.3390/network5010005.
5. Sharma SB, Bairwa AK. Leveraging AI for intrusion detection in IoT ecosystems: a comprehensive study. IEEE Access. 2025;13(6):66290–317. doi:10.1109/access.2025.3550392.
6. Alshamrani A, Myneni S, Chowdhary A, Huang D. A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities. IEEE Commun Surv Tutor. 2019;21(2):1851–77.
7. Handoyo Putro I, Ahmad T, Muslim Ijtihadie R. Enhancing MQTT intrusion detection in IoT using machine learning and feature engineering. IEEE Open J Commun Soc. 2025;6:7855–84. doi:10.1109/OJCOMS.2025.3610132.
8. Karimunda K, JdDM U, Bumbálek R, Zoubek T, Bartoš P, Kuneš R, et al. Machine learning-assisted cryptographic security: a novel ECC-ANN framework for MQTT-based IoT device communication. Computation. 2025;13(10):227. doi:10.3390/computation13100227.
9. Zeghida H, Boulaiche M, Chikh R. Securing MQTT protocol for IoT environment using IDS based on ensemble learning. Int J Inf Secur. 2023;22(4):1075–86. doi:10.1007/s10207-023-00681-3.
10. Ullah S, Ahmad J, Khan MA, Alshehri MS, Boulila W, Koubaa A, et al. TNN-IDS: transformer neural network-based intrusion detection system for MQTT-enabled IoT networks. Comput Netw. 2023;237(5):110072. doi:10.1016/j.comnet.2023.110072.
11. Karam K, Aqachtoul A, Elamrani A, Najib M, Rafalia N, Moumen I, et al. ISAAF: an IoT security and attack prevention framework using AI-driven predictive analytics. Sci Rep. 2025;15(1):44913. doi:10.1038/s41598-025-28516-2.
12. Allaga H, Biniz M, Farchane A. MQTTEEB-D: a high-fidelity benchmark for real-time MQTT anomaly detection using machine learning techniques. Ad Hoc Netw. 2025;181(22):104062. doi:10.1016/j.adhoc.2025.104062.
13. Qasem MA, Motiram BM, Thorat S, Al-Hejri AM, Alshamrani SS, Alshmrany KM, et al. Enhancement of cryptography algorithms for security of cloud-based IoT with machine learning models. Sci Rep. 2026;16(1):10972. doi:10.1038/s41598-026-45938-8.
14. Siddharthan H, Deepa T, Chandhar P. SENMQTT-SET: an intelligent intrusion detection in IoT-MQTT networks using ensemble multi cascade features. IEEE Access. 2022;10(3):33095–110. doi:10.1109/ACCESS.2022.3161566.
15. Santos L, Gonçalves R, Rabadão C, Martins J. A flow-based intrusion detection framework for internet of things networks. Cluster Comput. 2023;26(1):37–57. doi:10.1007/s10586-021-03238-y.

16. Schrötter M, Niemann A, Schnor B. A comparison of neural-network-based intrusion detection against signature-based detection in IoT networks. Information. 2024;15(3):164. doi:10.3390/info15030164.
17. Zeghida H, Boulaiche M, Chikh R, Bamhdi AM, Barros ALB, Zeghida D, et al. Enhancing IoT cyber attacks intrusion detection through GAN-based data augmentation and hybrid deep learning models for MQTT network protocol cyber attacks. Cluster Comput. 2025;28(1):58. doi:10.1007/s10586-024-04752-5.
18. Dandapat A, Mondal B. Design of intrusion detection system using GA and CNN for MQTT-based IoT networks. Wireless Pers Commun. 2024;134(4):2059–82. doi:10.1007/s11277-024-10984-w.
19. Mosaiyebzadeh F, Araujo Rodriguez LG, Macêdo Batista D, Hirata R. A network intrusion detection system using deep learning against MQTT attacks in IoT. In: Proceedings of the 2021 IEEE Latin-American Conference on Communications (LATINCOM); 2021 Nov 17–19; Santo Domingo, Dominican Republic. p. 1–6. doi:10.1109/LATINCOM53176.2021.9647850.
20. Khan MA, Khan MA, Jan SU, Ahmad J, Jamal SS, Shah AA, et al. A deep learning-based intrusion detection system for MQTT enabled IoT. Sensors. 2021;21(21):7016. doi:10.3390/s21217016.
21. Aqachtoul A, Karam K, Elamrani A, Najib M, Rafalia N, Bakhouya M. MQTTEEB-D: a real-world IoT cybersecurity dataset for AI-powered threat detection in MQTT networks. Mendeley Data. 2025:111897. doi:10.17632/jfttfjn6tr.1.