STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, December 2026, pp 5865–5879.
Published online in International Academic Press (www.IAPress.org)

Threat Detection in Internet of Things Messaging Protocol: Leveraging the MQTTEEB-D Realistic Dataset

Hamza ALLAGA [1], Mohamed BINIZ [1], Hayat SEMLALI [2], Abderrazak FARCHANE [1]

[1] *Sultan Moulay Slimane University, FPBM, Department of Informatics, LIMATI Laboratory, Beni Mellal, Morocco*
[2] *Cadi Ayyad University, FSTG, Department of Physics, LSEEET Laboratory, Marrakech, Morocco*

Abstract
The ubiquity of Internet of Things (IoT) devices has critically expanded the cyberattack surface, exposing inherent vulnerabilities in lightweight messaging protocols such as MQTT. This study rigorously evaluates machine learning (ML) techniques for threat detection using MQTTEEB-D, a dataset capturing diverse cyberattacks (DoS, MQTT Publish Flooding, SlowITe, Brute Force, Malformed Data) within a realistic IoT deployment. Our analysis reveals that a Random Forest (RF) classifier achieves, with 0.99 accuracy for binary (malicious vs. legitimate) classification and 0.97 accuracy for multi-class threat identification. In comparison, while the Multi-Layer Perceptron (MLP) achieves a notable 0.90 accuracy, it underperforms RF and struggles with high false-positive rates on legitimate traffic. Conversely, the Recurrent Neural Network (RNN) achieved only 0.74 accuracy, demonstrating that static packet features provide superior discriminative power compared to temporal dependencies for MQTT threat detection.These findings underscore the necessity of using realistic datasets to validate intrusion detection systems and highlight the efficacy of well-tuned ensemble models over more complex neural networks for securing MQTT-based IoT ecosystems.

Keywords IoT Cybersecurity, Threat Detection, Machine Learning (ML), Cyberattacks, MLP, Neural Networks,

DOI: 10.19139/soic-2310-5070-3031

1. Introduction

The proliferation of IoT has fundamentally reshaped the digital landscape, connecting billions of devices across critical sectors such as healthcare, smart cities, and industrial automation. This hyper-connectivity, however, simultaneously expands the cyberattack surface, exposing networks to a host of new threats. While the MQTT (Message Queuing Telemetry Transport) protocol is central to many IoT ecosystems and is favoured for its lightweight design and efficiency in resource-constrained environments [1]. However, these same characteristics—simplicity and minimal overhead—render MQTT networks inherently vulnerable to attacks ranging from denial-of-service to data breaches. Consequently, the development of robust, protocol-aware threat detection systems is no longer an ancillary measure but a critical requirement for protecting the security and integrity of IoT infrastructures [2].

Despite this pressing need, the effective detection of threats in realistic IoT environments faces significant challenges [3]. Traditional security solutions are often ill-suited to the unique traffic patterns and resource limitations of IoT devices [4]. However, a more fundamental obstacle is the scarcity of high-quality, publicly available datasets that accurately reflect contemporary attack vectors within specific protocols. This scarcity severely hinders the research and development of sophisticated ML-based Intrusion Detection Systems (IDS), as models trained on synthetic or outdated data often fail to generalise to live operational networks. This challenge is

*Correspondence to: Hamza ALLAGA (Email:hamzaallaga@gmail.com).

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2026 International Academic Press

particularly acute for MQTT security, where a nuanced understanding of both benign and malicious traffic patterns is essential for accurate classification.

To address these challenges, this study leverages the MQTTEEB-D dataset, which is a modern and realistic benchmark for evaluating threat detection mechanisms in MQTT networks. The primary objective of this study is to conduct a rigorous empirical analysis of ML methods for multiclass cyber-attack classification. Accordingly, the specific aims are threefold: 1) to evaluate and compare the performance of distinct model classes (ensemble methods and neural networks) in identifying diverse cyberattacks; 2) to assess model efficacy against specific threat types, including Bruteforce, DoS, Flood, Malformed packets, and SlowITe; and 3) to analyse their practical viability for distinguishing legitimate from malicious traffic, thereby informing their deployment potential.

In response to these challenges, this paper makes the following contributions:

1. **Evaluation on a contemporary, realistic dataset:** We conduct one of the first rigorous assessments of machine learning models on the MQTTEEB-D benchmark, which captures diverse realistic cyberattacks against a modern IoT testbed, addressing prior reliance on synthetic or outdated traffic.
2. **Computationally efficient modeling:** A Random Forest using 12 engineered features attains state-of-the-art accuracy, showing that high performance is achievable without complex architectures or large feature sets and enabling deployment on resource-constrained IoT gateways.
3. **Comparative performance baseline:** We establish a clear ranking (Random Forest $>$ Multi-Layer Perceptron $>$ RNN) for MQTT threat detection on MQTTEEB-D, providing actionable guidance on the strengths and limitations of each model class.
4. **Practical two-tier security architecture:** We propose a lightweight edge binary detector paired with a cloud-based multiclass forensic model, offering a deployable blueprint that secures IoT ecosystems without overburdening edge devices.

This paper is organized as follows. Section 1 establishes the IoT security context and analyzes MQTT protocol vulnerabilities. Section 2 reviews the existing literature on IoT security and intrusion detection systems. Section 3 details the experimental methodology, including the MQTTEEB-D dataset and model evaluation framework, while Section 4 specifies the experimental configuration and performance metrics. Section 5 presents the empirical results of the machine-learning models. These results are then thoroughly interpreted in Section 6, which includes a comparative analysis with state-of-the-art works, a discussion of practical implications, and an acknowledgement of the study's limitations. Finally, Section 7 summarises the key findings and outlines the future research directions.

2. Related Work

The security landscape has been fundamentally transformed by the swift growth of IoT devices, introducing significant vulnerabilities. IoT security is characterized by a persistent gap between technological innovation and adequate safeguard deployment, further complicated by device constraints including limited computational power and memory. These limitations impede the implementation of robust cryptographic mechanisms and facilitate the proliferation of threats, including botnets (for example, Mirai), Distributed Denial-of-Service (DDoS) attacks, Man-In-the-Middle (MitM) exploits, unauthorised access, and data breaches [6].

Conventional mitigation techniques, such as firewalls, are often ineffective in decentralised IoT environments [7]. Consequently, recent studies have increasingly concentrated on the creation of lightweight cryptographic protocols and the use of ML-based IDS for detecting anomalies in network traffic [8]. Reliable IDS require precise, protocol-specific behavioural models to accurately identify malicious activity.

The MQTT protocol is central to many IoT ecosystems, offering a lightweight publish–subscribe communication mechanism optimised for low-bandwidth and high-latency networks. Although MQTT's efficiency underpins its widespread adoption, its design priorities of simplicity and performance introduce significant security vulnerabilities. Standard MQTT implementations (v3.1.1) lack mandatory encryption, provide optional TLS support, and typically rely on basic username/password authentication, which is vulnerable to brute-force attacks [9].

Approaches that utilise ML and Deep Learning (DL) have shown potential in identifying complex network intrusions, such as MitM and DoSattacks [8]. However, existing studies have three primary limitations. First, most algorithm evaluations including those employing Support Vector Machines [10], a semi-supervised deep learning approach [11], and Deep Feed-Forward Neural Network [12] use datasets unrelated to MQTT (e.g., CIC-IDS2017, UNSW-NB15, KDD-99) [12, 13], which limits protocol-specific insights. Second, MQTT-focused research typically addresses a small subset of attack types, often fewer than five [14, 15, 16], thereby restricting the diversity of threat coverage. Third, validation practices are frequently inadequate; some studies lack detailed attack descriptions [17], whereas others rely on outdated benchmarks [13]. The introduction of the MQTTset dataset by Vaccari et al. [18] has mitigated some data scarcity; however, challenges remain owing to the computational demands of DL methods [19] and the limited scope of attack scenarios, which hinder model generalisability [20, 16].

A major obstacle to advancing IoT security research is the lack of high-fidelity publicly available datasets. Historically, the field has depended on benchmark datasets such as KDD-99 and NSL-KDD, which are now considered obsolete and unrepresentative of modern network traffic or attack vectors, especially in IoT contexts [21]. More recent datasets, such as CIC-IDS2017 and Bot-IoT, have improved realism and relevance by incorporating contemporary attack scenarios [22, 23].

However, significant limitations persist. Many available datasets are synthetic, reducing their realism, or lack protocol-specific features necessary for specialised detection models. In particular, few public datasets offer comprehensive labelled collections of MQTT-specific attacks. This gap is critical because the effectiveness of ML models is closely tied to the quality and relevance of their training data. Models trained on unrealistic data tend to exhibit poor generalisation and high false-positive rates during deployment, diminishing their practical utility.

A critical analysis of the literature reveals a clear scarcity of empirical studies that evaluate modern ML models using publicly available, realistic datasets. This issue is particularly acute for security solutions that must be both MQTT-specific and robust against a diverse array of contemporary attack vectors. Our study directly addresses this scarcity by leveraging the MQTTEEB-D dataset. Thus, we provide a rigorous and reproducible performance benchmark that is essential for advancing the development of more practical and effective intrusion detection systems for realistic IoT ecosystems.

3. Experimental Methodology

This section details the experimental methodology designed to evaluate the efficacy of the ML technique for threat detection in MQTT-based IoT networks. The overall research framework, from data acquisition to model evaluation, is illustrated in Figure 1.

3.1. MQTTEEB-D Dataset

To address the critical scarcity of high-fidelity public datasets, this study utilizes MQTTEEB-D [24], a benchmark generated from a physical IoT testbed. To position this dataset within the existing literature, Table 1 provides a comparative analysis against other prominent MQTT-focused datasets, evaluating them based on testbed fidelity.

Table 1. Comparative Analysis of MQTT Cybersecurity Datasets.

Dataset	Testbed Type	Hardware	Sensor Source	Multi-Attack
MQTT-IoT-IDS2020 [15]	Simulated	None (Software)	Simulated	Yes
MQTTset [18]	Emulated	None (IoT-Flock)	Emulated	Yes
SENMQTT-SET [25]*	Hybrid	Physical (RPi)	Simulated	No
MQTTEEB-D (This Study)	**Fully Physical**	**Physical (MySignals)**	**Physical (Sensors)**	**Yes**

*Note: Please add the full reference for SENMQTT-SET to your bibliography.

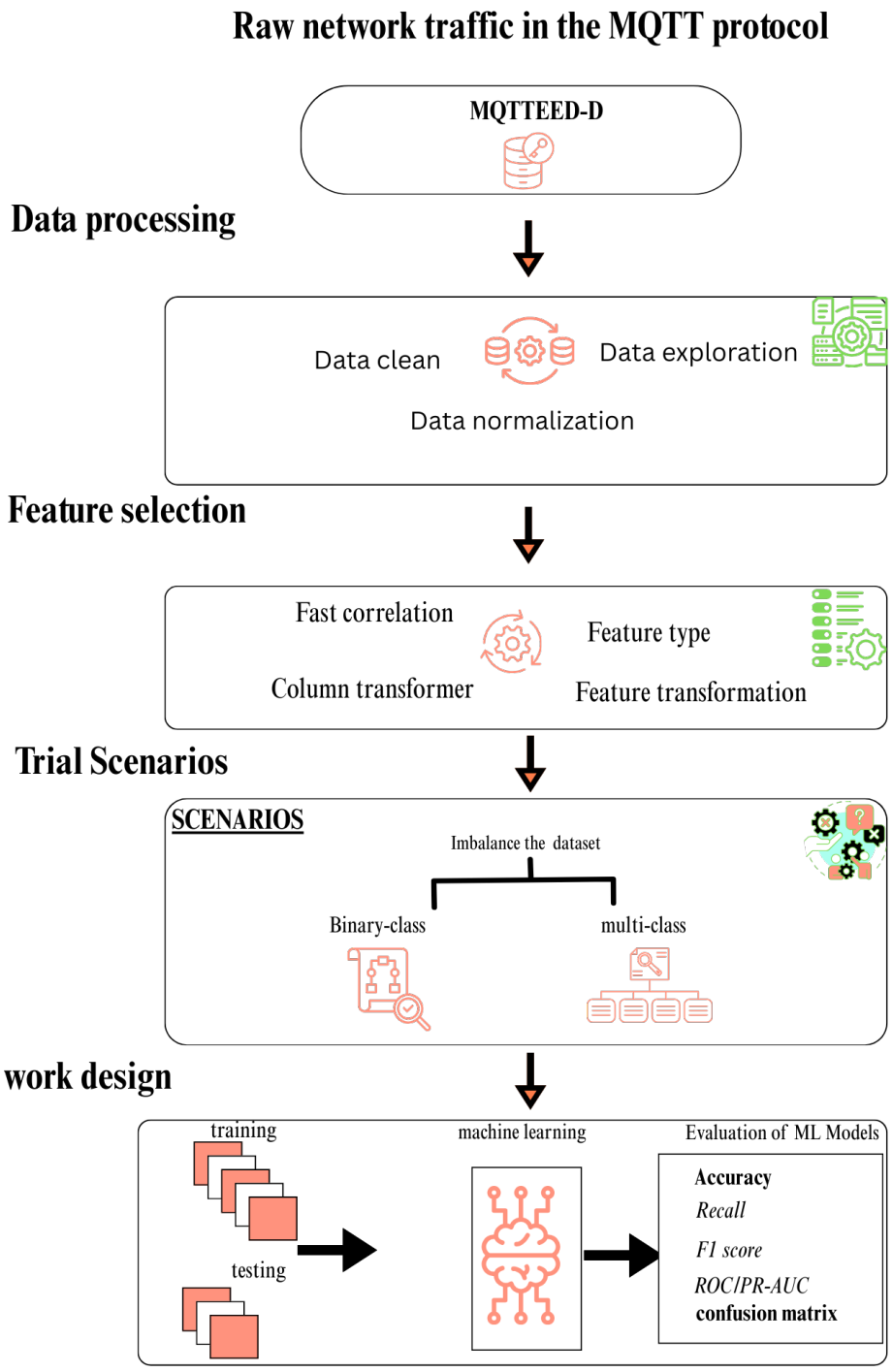


Figure 1. Experimental workflow for IoT threat detection, illustrating data acquisition, preprocessing, feature engineering, model training, and performance evaluation stages.

The comparison in Table 1 highlights a clear progression in data fidelity, establishing MQTTEEB-D as a state-of-the-art benchmark for evaluating security mechanisms in realistic IoT environments.

The dataset was generated from this real-time testbed at the International University of Rabat, which uses MySignals IoT health sensors and Raspberry Pi 4 devices as gateways communicating with a central MQTT broker. This physical setup represents the actual complexity of active IoT communication that synthetic data fails to offer. To bridge the gap between simulated and real-world scenarios, a diverse set of cyberattacks including Denial of Service (DoS), SlowITe, Malformed Data injection, Brute Force, and MQTT publish flooding were executed in real-time.

Network traffic was captured using a Python wrapper for tshark (PyShark) and systematically organized into CSV files. To ensure high data quality for model training, the raw data underwent a rigorous preprocessing pipeline that included outlier removal, feature normalization, and standardization. The resulting dataset provides the foundation for the feature engineering process detailed in the following section.

3.2. Feature Engineering

This section details the feature engineering process for machine learning-based network threat detection. From an initial dataset comprising 222,813 network traffic instances, a candidate set of features was extracted based on domain knowledge of the TCP/IP and MQTT protocols. To ensure that only the most discriminative variables were retained, we conducted a comprehensive feature importance analysis using multiple statistical and model-based methods (including F-score, Mutual Information, and RF Importance). A composite score reflecting each feature's overall importance was calculated and is presented in the final column of Table 2. The final set of 11 explanatory variables represents the features with the highest discriminative power according to this analysis. These features create a multifaceted representation of network communications by capturing semantics from both the transport and application layers, and a comprehensive description of each is provided in Table 2.

Transport layer features were selected to model the fundamental TCP/IP connection characteristics. These include the TCP connection state (tcp_flags), interpacket temporal dynamics (tcp_time_delta), and packet payload size (tcp_len), providing a low-level view of the traffic flow.

Application-layer features were extracted through deep packet inspection of the Message Queuing Telemetry Transport (MQTT) protocol. This process yielded variables that characterised critical aspects of the protocol behaviour, including session management (e.g. mqtt_conack_flags), message delivery guarantees (mqtt_qos), and header configurations (mqtt_hdrflags).

The resulting dataset presented three primary preprocessing challenges. First, a significant class imbalance was observed, with the malformed-attack class constituting 46.8% of the labelled instances. Second, the feature set contained a mixture of data types—continuous, categorical, and binary—which necessitated consistent encoding schemes. Third, the wide variance in the scales of the continuous features required normalisation to ensure equitable contribution during model training. The final engineered feature vector effectively integrates low-level network metrics with high-level protocol semantics, providing a discriminative basis for identifying the malicious traffic patterns.

3.3. Experimental Design and Model Evaluation

Our experimental design was structured to assess model performance across two distinct classification tasks: binary anomaly detection and multiclass threat identification.

3.3.1. Model Selection Three ML methods were selected to provide a comparative analysis across different algorithmic paradigms:

Random Forest (RF): An ensemble method chosen for its established robustness, high accuracy, and ability to handle high-dimensional data without extensive tuning.

Multi-Layer Perceptron (MLP): A foundational feedforward neural network selected as a baseline for deep learning performance on structured data.

Recurrent Neural Network (RNN): This model was selected to assess its capability in capturing temporal dependencies within network traffic sequences.

Table 2. Description and Importance of Engineered Features

Feature Name	Type (Details)	Description	Importance Score
Transport Layer			
`tcp_flags`	Categorical (3 values)	Bitmap indicating TCP connection state.	1.80
`tcp_time_delta`	Continuous	Time elapsed since the previous packet.	4.61
`tcp_len`	Integer [1–3,936]	Length of the TCP segment payload in bytes.	3.80
Application Layer (MQTT)			
`mqtt_conack_flags`	Mixed (19 values)	Connection acknowledgment status flags.	0.01
`mqtt_conflag_cleansess`	Binary [0, 1]	Flag for requesting a new client session.	0.49
`mqtt_conflags`	Mixed (4 values)	Composite bitmap of connection request flags.	0.84
`mqtt_dupflag`	Binary [0, 1]	Flag for a re-transmitted message.	0.17
`mqtt_hdrflags`	Categorical (44 values)	Header defining message type and QoS.	1.83
`mqtt_kalive`	Integer [0–60]	Client-defined keep-alive interval (s).	0.58
`mqtt_msg`	Integer [0–65,504]	MQTT message payload size in bytes.	2.28
`mqtt_qos`	Ordinal [0–3]	Quality of Service level for the message.	1.05

3.3.2. Classification Scenarios The models were evaluated under the following two scenarios:

- **Binary Classification:** This scenario focused on the core task of anomaly detection. The objective was to classify all network traffic into one of two classes: Benign or Malicious. The Malicious class aggregated all recorded attack types.
- **Multiclass Classification:** This scenario addressed fine-grained threat identification, requiring the model to classify traffic into six distinct classes: Benign, DoS, SlowITe, Malformed, Brute Force, and Flood.

subsubsectionImplementation and Training: In both cases, the MQTTEEB-D dataset was divided into training (80%) and testing (20%) subsets, utilizing stratified sampling to preserve the class distribution across these divisions. To directly evaluate the models' robustness against the dataset's natural traffic distribution, no explicit class imbalance techniques, such as resampling (e.g., SMOTE) or class weighting, were applied. Model training was performed on the training set, with hyperparameter optimization conducted using a grid search methodology. The final performance metrics were reported based on the models' predictions on the unseen testing set.

4. Experimental Setup and Evaluation

4.1. Work Environment

The research was performed using a Lenovo ThinkPad L13 Gen 1 notebook featuring an Intel® Core™ i5-10310U processor (operating at 1.70 GHz with 8 cores), 8 GB of RAM memory, and Intel® UHD Graphics (CML GT2). The machine operates with a 256.1 GB SSD storage drive and operates on Ubuntu 22.04.2 LTS. For the Cloud-related experiments, Google Colab served as our computational platform. Google Colab provides 12.67 GB of total RAM capacity (with 11.46 GB accessible for use), a dual-core processor, and 112.6 GB of storage space, in addition to providing access to NVIDIA K80 GPU units.

4.2. Tools and Libraries

The trials were carried out using Python (version 3.10.4) along with several frameworks: Pandas (1.4.2) for data handling and coding, Scikit-learn (1.1.1) for implementing the models ML and preprocessing, TensorFlow (2.9.1) for developing neural networks, Matplotlib for data visualization, and Google Colab for cloud-based execution and storage.

4.3. Performance Evaluation Metrics

Our approach utilizes multiple assessment criteria to measure the effectiveness of the developed model, encompassing accuracy, precision, recall, F1-score, ROC curve (Receiver Operating Characteristic curve) analysis, and confusion matrix evaluation. The subsequent explanations detail these evaluation metrics:

4.3.1. Confusion Matrix The **confusion matrix** serves as a fundamental instrument for assessing the effectiveness of ML classification algorithms. It displays a tabular format presenting four possible outcomes derived from predicted and ground truth data: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [26]. As demonstrated in Table 3, TP denotes correctly identified positive instances, TN signifies properly classified negative instances,FP represents misclassified positive predictions, and FN indicates incorrectly classified negative instances. This evaluation framework is vital for computing essential efficiency indicators, including Recall, F1-score, Accuracy, and Precision.

Table 3. Structure of a Standard Confusion Matrix for Binary Classification

	Actual Positive	**Actual Negative**
Predicted Positive	**TP**	**FP**
Predicted Negative	**FN**	**TN**

4.3.2. Classifier Evaluation Using AUC and ROC Analysis The **ROC curve** serves as a fundamental instrument for evaluating the effectiveness of classification models [29]. It depicts the relationship between sensitivity (true positive rate) and specificity (1 - false positive rate) across various decision boundaries, enabling practitioners to choose classification algorithms that match different error tolerance requirements and precision objectives [27].

The AUC (Area Under Curve) measures a model's ability to distinguish between classes by calculating the area under the ROC curve. An AUC value approaching 1.0 signifies a robust classification capability, while a value of 0.5 indicates performance no better than random chance. The true positive rate is represented on the Y-axis, whereas the X-axis shows the false-positive rate. This graphical representation illustrates classification effectiveness, as documented by [28].

Table 4. Evaluation Metrics for Attack Classification

Metric	Description	Mathematical Formulation
Accuracy	Proportion of all instances correctly classified; can be misleading under class imbalance.	$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$
Precision	Proportion of predicted positives that are true positives (positive predictive value).	$\text{Precision} = \frac{TP}{TP+FP}$
Recall	Proportion of actual positives correctly identified (sensitivity).	$\text{Recall} = \frac{TP}{TP+FN}$
F1-score	Harmonic mean of precision and recall; summarizes their trade-off.	$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$

Note: TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives.

5. Analysis of Experimental Results

This section evaluates three ML models RF, MLP, and RNN using the MQTTEEB-D dataset. The models were assessed for their capacity for multiclass threat classification, with a focus on performance across imbalanced classes and the ability to distinguish legitimate traffic from malicious activity. The aggregate performance metrics, summarized in Table 5, establish a clear performance hierarchy, where the RF model substantially outperforms the neural network architectures. RF significantly outperformed the other models with 0.97 accuracy and 0.96-0.97 F1-scores, demonstrating superior performance across all metrics. The performance hierarchy shows RF $>$ MLP (0.91 accuracy) $>$ RNN (0.75 accuracy), suggesting that traditional ensemble methods excel over deep learning approaches for this particular classification task, possibly because of dataset characteristics or feature engineering advantages.

Table 5. Detailed Performance Comparison of All Classification Models Across All Traffic Classes. The F1-Score is highlighted in bold for the best-performing model in each class row. All metrics are presented as decimals. Total support is 44,514 samples.

	Random Forest (RF)			**Multi-Layer Perceptron (MLP)**			**Recurrent Neural Network (RNN)**		
Attack	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Bruteforce	0.99	0.99	**0.99**	0.76	0.96	0.85	0.74	0.73	0.73
DoS	0.97	0.97	**0.97**	0.95	0.98	**0.97**	0.80	0.58	0.68
Flood	0.98	0.98	**0.98**	0.99	0.95	0.97	0.99	0.94	0.97
Legitimate	0.91	0.85	**0.88**	0.55	0.18	0.28	0.58	0.16	0.25
Malformed	0.98	0.99	**0.98**	0.94	1.00	0.97	0.72	0.96	0.82
SlowITe	0.97	0.95	**0.96**	0.97	0.85	0.91	0.87	0.41	0.56
Overall Model Performance									
Accuracy		0.970			0.909			0.748	
Macro Avg F1-Score		0.96			0.82			0.67	
Weighted Avg F1-Score		0.97			0.89			0.72	

5.1. Random Forest: A Robust Baseline

The RF model demonstrated exceptional performance. A granular analysis of its multiclass confusion matrix (Figure 2) reveals high true positive rates for malformed (20,914), DoS(8,134), and brute force (5,395) attacks.

This indicates that their feature patterns are highly discriminative. The model's primary limitation was a modest confusion involving the legitimate class, which was most frequently misclassified as malformed (305 instances),

Crucially, the model also demonstrated high computational efficiency, with an average prediction time of only 0.1804 seconds across the entire test set, validating its potential for low-latency environments.

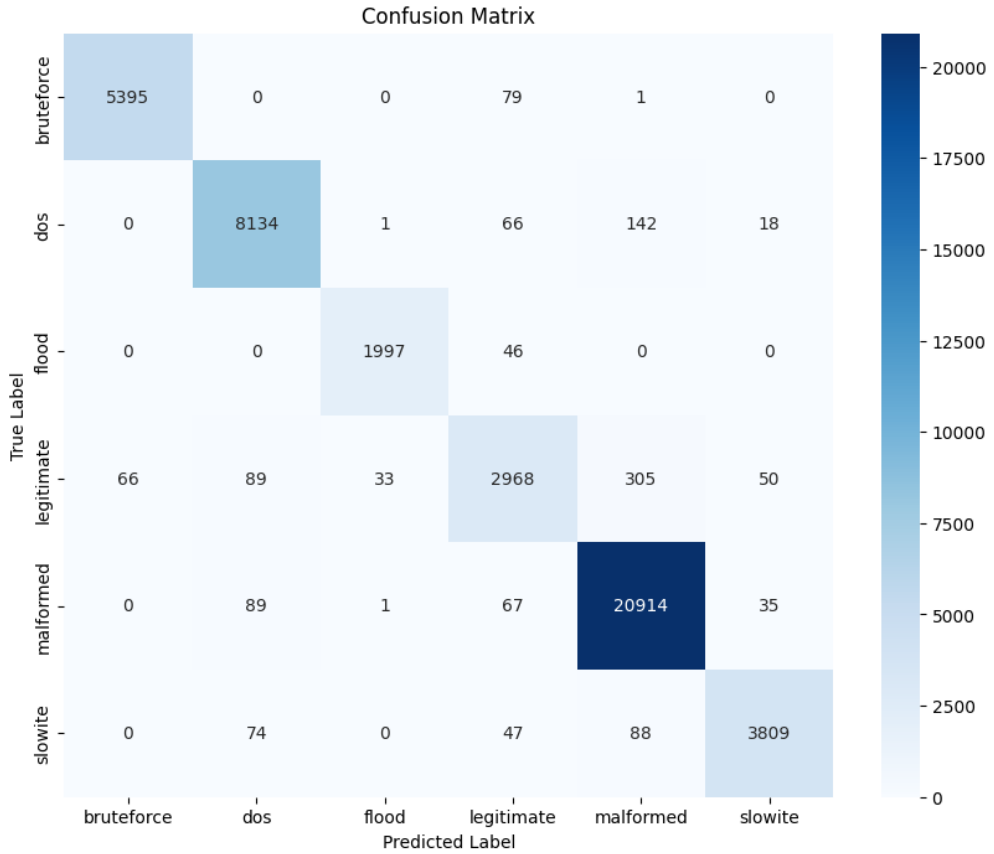


Figure 2. Confusion matrix for the RF classifier. Diagonal cells indicate high true positive rates for attack classes, while off-diagonal values highlight the misclassification of legitimate traffic as the primary source of error.

To further assess its core anomaly detection capability, a binary classification was performed (Table 6). In this task, the RF model achieved near-perfect accuracy (0.99) and, critically, a recall of 1.00 for the malicious class. This perfect sensitivity, complemented by a precision of 0.99, confirms the model's suitability as a robust tool for first-pass threat detection, as the features separating any malicious activity from legitimate traffic are highly distinct.

Table 6. Binary Classification Report for the Random Forest (RF) Model.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Legitimate Traffic	1.00	0.99	0.99	23,457
Malicious Traffic	0.99	1.00	0.99	21,106
Accuracy			**0.99**	**44,563**

5.2. Neural Network Performance: MLP and RNN

The performance of the neural network models was evaluated to provide a comparative analysis against the ensemble-based RF.

5.2.1. Multi-Layer Perceptron MLP model demonstrated a bifurcated performance profile: while it was highly effective at identifying all five classes of attack traffic, it struggled significantly to correctly classify legitimate network behaviour. The model achieved an overall test accuracy of 0.909; however, as detailed in the comprehensive performance summary in Table 5, this aggregate score masks a critical underlying weakness.

The MLP achieved high F1-scores for most attack types, including DoS(0.97), Flood (0.97), and SlowITe (0.91). This indicates that the neural network successfully learned the distinct feature patterns associated with these activities. However, its performance on the legitimate class was exceptionally poor, with an F1-score of only 0.28, driven by a very low recall of 0.18. This indicates that the model correctly identified only 18% of all legitimate traffic instances, misclassifying the vast majority as attacks. This high false-positive rate renders the model impractical for realistic deployment, as it generates an unmanageable volume of false alarms.

The MLP elevated false-positive rate is primarily attributable to its susceptibility to class imbalance, a vulnerability compounded by the moderate size of the training dataset. During optimization, a conventional cross-entropy loss function causes the training process to be disproportionately influenced by the over-represented and more distinct attack classes, such as Malformed and DoS traffic. The limited number of 'legitimate' instances is consequently insufficient for a data-intensive architecture like the MLP to construct a robust and generalizable model of benign behavior. This deficiency results in the failure to establish a nuanced decision boundary, leading to a systemic bias where ambiguous instances are classified as malicious. In contrast, the ensemble structure of the Random Forest model demonstrates superior inherent resilience to these combined challenges, enabling it to effectively model both majority and minority classes..

This performance dichotomy is visually corroborated by the ROC analysis in Figure 3. The Area Under the Curve (AUC) values for all five attack classes were excellent, consistently exceeding 0.98, indicating strong discriminative power. In stark contrast, the legitimate class was a clear outlier, with a significantly lower AUC of 0.862. The deviation of its curve toward the random classifier baseline visually confirms the difficulty in distinguishing benign patterns from malicious ones. Consequently, while the MLP shows promise for attack signature recognition, its utility is severely limited by its inability to reliably model normal network traffic.

5.2.2. Recurrent Neural Network evaluation involved employing an input data structure consistent with that used for the Random Forest and Multi-Layer Perceptron models. Specifically, the network was provisioned with a two-dimensional feature matrix, wherein each row encapsulated the feature vector of a single, independent packet. This methodology was conceived to ascertain the RNN architecture's capacity for implicitly discerning temporal relationships from intra-vector features, such as tcp_time_delta, thereby obviating the necessity for explicitly constructed input sequences. However, despite a rigorous hyperparameter optimization process, the resultant model failed to demonstrate competitively. It is important to note that a simple RNN architecture was chosen to establish a baseline for temporal analysis; more complex variants like LSTMs or GRUs might yield different results.

Among the assessed architectures, the RNN model demonstrated the poorest performance, underscoring a fundamental disconnect between its sequential processing abilities and the characteristics of the dataset features. With an overall accuracy of only 0.748, the RNN results were highly erratic and significantly trailed behind both the RF and MLP models.

A detailed review of its per-class performance in Table 5 reveals this inconsistency in the results. Although the model demonstrated surprising efficacy in detecting Flood attacks with a high F1-score of 0.97, its capacity to identify other critical threats was severely compromised. It struggled significantly with DoS(recall of 0.58) and SlowITe (recall of 0.41), indicating that it could not detect a large proportion of these types of attacks. Mirroring the MLP, its most critical failure was on legitimate traffic, where an F1-score of just 0.25 confirms its impracticality for any realistic deployment owing to the overwhelming rate of false positives. This consistent underperformance reveals a critical insight: treating network packets as discrete, independent instances negates the primary architectural advantage of RNNs—their capacity for sequential data processing. The literature

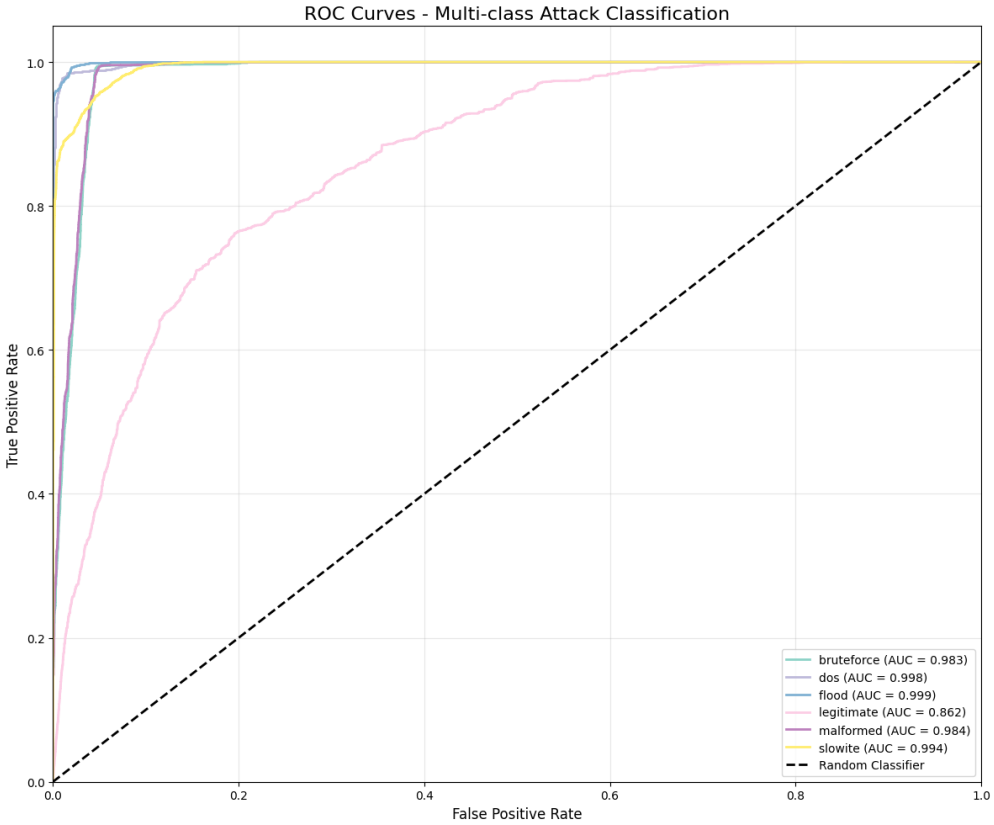


Figure 3. ROC curves for the MLP classifier, illustrating high discriminative power for all attack classes (AUC $>$ 0.98) and a comparatively weaker performance on the legitimate class (AUC = 0.862).

demonstrates that the inclusion of time-based features within a static vector is insufficient for the network to implicitly infer meaningful temporal dependencies. Consequently, the effective deployment of RNN architectures for threat detection appears to be predicated on the explicit formulation of input data into temporal sequences. RNNs are architecturally designed to capture long-range dependencies in sequences; however, the results imply that such dependencies were either not present or not informative for this classification task. The model's inherent complexity, geared towards time-series analysis, appears to have become a liability, preventing it from effectively learning the more direct, non-sequential patterns that the RF and MLP models successfully captured.

This interpretation is visually reinforced by the ROC analysis shown in Figure 4. The widely dispersed curves, with DoS(AUC=0.815) and Legitimate (AUC=0.841) performing near the lower bounds, illustrate the model's general struggle to establish reliable decision boundaries. Ultimately, the RNN architecture, as implemented here, proved ill-suited for this problem, demonstrating that applying complex deep learning models without a corresponding alignment in feature representation offers no advantage and can even degrade performance.

6. Discussion

The experimental results demonstrate clear performance differences among ML paradigms for MQTT threat detection. This section interprets these findings, compares them with state-of-the-art works, discusses practical implications, and acknowledges study limitations.

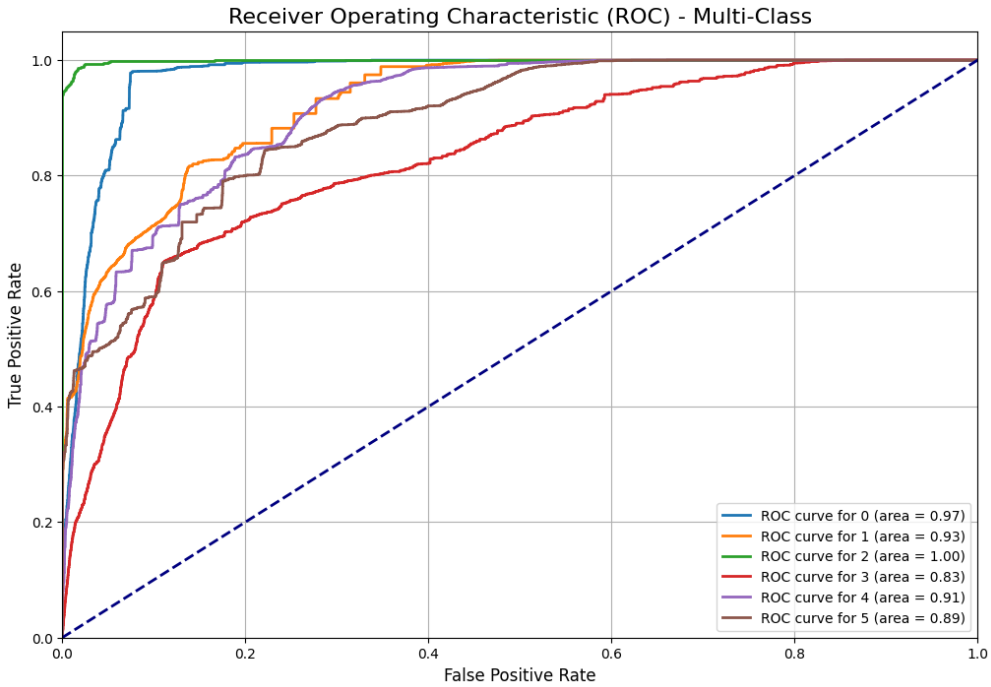


Figure 4. ROC curves for the RNN classifier, showing varied performance across classes, with Flood (AUC=0.997) being the most accurately classified and DoS(AUC=0.815) the least.

6.1. Interpretation of Model Performance

The outstanding performance of the RF model, which achieved 0.97 multiclass accuracy, underscores a critical insight: for structured, tabular network data, the quality of feature engineering can be more decisive than the complexity of the model architecture. The MQTTEEB-D dataset provides well-defined engineered features, and RF excels in this environment because its core mechanism is exceptionally effective at identifying complex decision boundaries within such a feature space. The model's high true positive rates for malformed, DoS, and bruteforce attacks suggest that these threats introduce statistically significant and consistent deviations in the feature set, which the RF can easily isolate.

In contrast, the neural network models, particularly RNN, did not yield commensurate performance improvements. The strong but inferior performance of the MLP suggests that its layered transformations did not uncover patterns deeper than those already captured by the ensemble method. The weaker performance of the RNN model (0.74 accuracy) was particularly revealing. This implies that for this dataset, the temporal or sequential relationships between individual network packets were less informative than the static per-packet features. This suggests that the attacks captured in MQTTEEB-D are better characterised by the instantaneous properties of their traffic than by their evolution over time.

6.2. Comparative Analysis with State-of-the-Art

To rigorously evaluate the contribution of our work, we provide a comparative analysis of prominent state-of-the-art studies on MQTT intrusion detection. A summary of this comparison is presented in Table 7.

The analysis presented in Table 7 positions our work favourably in the current research landscape. A significant advantage of our approach is its **computational efficiency**, as our models achieved high performance using a concise set of only **12 features**. This is in stark contrast to related studies, which utilised feature sets ranging from 29 to 52 features. This efficiency is critical for deployment on resource-constrained IoT Gateway.

Table 7. Performance Comparison with State-of-the-Art MQTT Intrusion Detection Systems. Note the high efficiency of our model, which uses only 12 features.

Study	Dataset & Features	Model	Accuracy	Precision	Recall	F1-Score
[20] (Binary)	Packet-flow (52 features)	RF	0.8999	0.9035	0.8898	0.8966
[20] (Multi)	Bi-flow (29 features)	RF	0.9798	0.9639	0.8437	0.8998
[15] (Multi)	MQTT-IoT-IDS2020 (33 features)	RF	0.9160	-	-	0.9140
[15] (Multi)	MQTT-IoT-IDS2020 (33 features)	MLP	0.9039	-	-	0.9019
This Study (Binary)	**MQTTEEB-D (12 features)**	**RF**	**0.9926**	**0.9900**	**1.0000**	**0.9900**
This Study (Multiclass)	**MQTTEEB-D (12 features)**	**RF**	**0.9709**	**0.9600**	**0.9600**	**0.9600**

In the **binary classification** task, our RF model achieved an accuracy of **0.99**, substantially outperforming the 0.89 reported by Khan et al. [20]. More critically, our model achieved a perfect recall of **1.00** for the malicious class, which is a paramount metric for any security system. In the more challenging **multiclass classification** scenario, our model's accuracy of **0.97** is highly competitive and notably higher than the 0.91 achieved by Hindy et al. [15]. Although Khan et al. [20] reported a slightly higher accuracy, their model exhibited a significantly lower recall (0.84), suggesting a potential weakness in detecting certain attacks. Our model demonstrates a more balanced and robust performance while maintaining superior efficiency.

6.3. Practical Implications and Tiered Architecture

A key takeaway from our results is a practical path forward for securing resource-constrained IoT ecosystems. A monolithic, resource-intensive multiclass model on each edge device is often impractical. Instead, our findings strongly advocate for a **tiered security architecture**.

1. **Tier 1 (Edge Detection):** A lightweight, computationally efficient binary classifier, such as the one developed in this study, could be deployed on IoT gateways. With its 0.99 accuracy and perfect recall for malicious traffic, and a prediction latency of just 0.1804 seconds, this model acts as a highly reliable first-pass filter, ensuring that no threats are missed while demanding minimal computational overhead.
2. **Tier 2 (Cloud/Server Forensics):** Traffic flagged as suspicious by the Tier 1 filter would be escalated to a more powerful, centralized server. This server runs a comprehensive multiclass model (such as our multiclass RF) for in-depth forensic analysis to identify the specific attack type.

This hybrid approach optimises both the detection speed and analytical depth, providing robust security without overburdening the edge devices.

The practical deployment of such a tiered architecture necessitates addressing several critical challenges. First, communication overhead between the edge and cloud tiers must be carefully managed to prevent network bottlenecks. This is typically achieved by having the edge device transmit concise metadata or feature vectors of suspicious flows, rather than raw packet data, thereby minimizing data volume. Second, the communication channel itself must be secured, for instance via a Virtual Private Network (VPN) or Transport Layer Security (TLS), to ensure the integrity and confidentiality of the transmitted data and to prevent the channel from becoming an attack vector. Finally, the end-to-end latency—from packet ingress at the edge to final classification in the cloud—must be acceptable for the intended security application. Although this two-step process is unsuitable for attack mitigation requiring microsecond-level response times, it is highly effective for forensic analysis and the identification of compromised devices for subsequent remediation, aligning with the primary objectives of an intrusion detection system.

6.4. Study Limitations

To accurately interpret the results of this study, it is important to acknowledge its limitations. First, the MQTTEEB-D dataset, despite its realism, originates from a specific physical testbed that emulates a healthy IoT environment

in controlled laboratory conditions. Consequently, the generalizability of these findings to other IoT domains, such as industrial or smart home settings with disparate network topologies and traffic patterns, requires validation. Similarly, claims of computational efficiency are based on performance on a standard computer; validation on a representative edge device, such as a Raspberry Pi, is required to confirm real-world deployment viability. Furthermore, the models were trained and evaluated exclusively on the five attack types represented in the dataset; their performance against novel or zero-day threats in a dynamic threat landscape has not been determined.

7. Conclusion

This study confirmed the superior efficacy of a RF model for multiclass threat classification, which achieved 0.97 accuracy on the MQTTEEB-D dataset and substantially outperformed the MLP and RNN architectures. Its principal limitation, a challenge shared by all evaluated models, was the misclassification of legitimate traffic, which contrasts with the near-perfect performance observed in the simpler binary detection task. This study underscores a key insight for IoT security: for structured data, algorithmic selection and feature engineering are often more critical than architectural complexity.

The findings directly inform practical security deployments. We recommend a tiered intrusion detection architecture: a computationally efficient binary classifier acts as a high-confidence, first-pass filter, whereas flagged events are escalated to a more resource-intensive multiclass model for forensic analysis. This approach optimises both the detection speed and analytical depth. Consequently, practitioners should prioritise the robust modelling of normal traffic patterns to minimise the operational burden of false positives.

The weaker performance of RNN suggests that while standard temporal models may be insufficient, more sophisticated architectures are warranted. Therefore, a key future direction is the application of advanced sequence models, such as Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), and Transformers, to more robustly investigate and capture complex temporal dependencies in network traffic. This pursuit must be balanced with the evaluation of the computational overhead of these models to ensure their viability for deployment on resource-constrained IoT devices.

REFERENCES

1. M. Husnain, K. Hayat, E. Cambiaso, U. U. Fayyaz, M. Mongelli, H. Akram, S. Ghazanfar Abbas, and G. A. Shah, *Preventing MQTT vulnerabilities using IoT-enabled intrusion detection system*, Sensors, vol. 22, no. 2, p. 567, 2022.
2. J. Roldán-Gómez, J. Carrillo-Mondéjar, J. M. Castelo Gómez, and S. Ruiz-Villafranca, *Security analysis of the MQTT-SN protocol for the Internet of Things*, Applied Sciences, vol. 12, no. 21, p. 10991, 2022.
3. D. Canavese, L. Mannella, L. Regano, and C. Basile, *Security at the edge for resource-limited IoT devices*, Sensors, vol. 24, no. 2, p. 590, 2024.
4. B. R. Kikissagbe, and M. Adda, *Machine learning-based intrusion detection methods in IoT systems: A comprehensive review*, Electronics, vol. 13, no. 18, p. 3601, 2024.
5. M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, *A survey of machine and deep learning methods for Internet of Things (IoT) security*, IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1646–1685, 2020.
6. M. Antonakakis, et al., *Understanding the Mirai botnet*, in 26th USENIX Security Symposium (USENIX Security 17), Vancouver, BC, pp. 1093–1110, 2017.
7. S. Ullah, et al., *TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT networks*, Computer Networks, vol. 237, p. 110072, 2023.
8. N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, *Network intrusion detection for IoT security based on learning techniques*, IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2671–2701, 2019.
9. A. Bashir, and A. H. Mir, *Lightweight Secure-MQTT for Internet of Things*, in Internet of Things (IoT), Springer, Singapore, pp. 57–66, 2020.
10. M. Mohammadi, et al., *A comprehensive survey and taxonomy of the SVM-based intrusion detection systems*, Journal of Network and Computer Applications, vol. 178, p. 102983, 2021.
11. M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, and M. J. Ryan, *Semi-supervised spatio-temporal deep learning for intrusions detection in IoT networks*, IEEE Internet of Things Journal, vol. 8, no. 15, pp. 12251–12265, 2021.
12. O. Faker, and E. Dogdu, *Intrusion detection using big data and deep learning techniques*, in Proceedings of the ACM Southeast Conference, Kennesaw, GA, USA, pp. 86–93, 2019.
13. A. Shalaginov, O. Semeniuta, and M. Alazab, *MEML: Resource-aware MQTT-based machine learning for network attacks detection on IoT edge devices*, in Proceedings of the IEEE/ACM International Conference on Utility and Cloud Computing Companion, Auckland, New Zealand, pp. 123–128, 2019.

14. H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Munoz-Castaneda, I. Garcia, and C. Benavides, *Multiclass classification procedure for detecting attacks on MQTT-IoT protocol*, Complexity, vol. 2019, Article ID 6516253, 2019.
15. H. Hindy, D. Brosset, E. Bures, A. Tachtatzis, R. Atkinson, and X. Bellekens, *Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 dataset)*, in Proceedings of the International Network Conference, Cham: Springer, pp. 73–84, 2020.
16. I. Ullah, A. Ullah, and M. Sajjad, *Towards a hybrid deep learning model for anomalous activities detection in Internet of Things networks*, IoT, vol. 2, no. 3, pp. 428–448, 2021.
17. E. Ciklabakkal, A. Dönmez, M. Erdemir, E. Süren, M. K. Yilmaz, and P. Angin, *ARTEMIS: An intrusion detection system for MQTT attacks in Internet of Things*, in Proceedings of the IEEE 38th Symposium on Reliable Distributed Systems, Lyon, France, pp. 369–370, 2019.
18. I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, *MQTTset, a new dataset for machine learning techniques on MQTT*, Sensors, vol. 20, no. 22, p. 6578, 2020.
19. R. Vinayakumar, et al., *Deep learning approach for intelligent intrusion detection system*, IEEE Access, vol. 7, pp. 41525–41550, 2019.
20. M. A. Khan, M. A. Khan, S. U. Jan, J. Ahmad, S. S. Jamal, A. A. Shah, N. Pitropakis, and W. J. Buchanan, *A deep learning-based intrusion detection system for MQTT enabled IoT*, Sensors, vol. 21, no. 21, p. 7016, 2021.
21. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, *A detailed analysis of the KDD CUP 99 data set*, in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, pp. 1–8, 2009.
22. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, *Toward generating a new intrusion detection dataset and intrusion traffic characterization*, in Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), Funchal, Madeira, Portugal, pp. 108–116, 2018.
23. N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Turnbull, *Towards the development of a realistic intrusion detection dataset: The Bot-IoT dataset*, Future Generation Computer Systems, vol. 100, pp. 779–796, 2019.
24. A. Aqachtoul, K. Karam, A. Elamrani, M. Najib, N. Rafalia, and M. Bakhouya, *MQTTEEB-D: A real-world IoT cybersecurity dataset for AI-powered threat detection in MQTT networks*, Mendeley Data, V1, 2025.
25. H. Siddharthan, T. Deepa and P. Chandhar, "SENMQTT-SET: An Intelligent Intrusion Detection in IoT-MQTT Networks Using Ensemble Multi Cascade Features," in IEEE Access, vol. 10, pp. 33095-33110, 2022, doi: 10.1109/ACCESS.2022.3161566.
26. M. A. Talukder, M. M. Islam, M. A. Uddin, K. F. Hasan, S. Sharmin, S. A. Alyami, and M. A. Moni, *Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction*, Journal of Big Data, vol. 11, no. 1, p. 33, 2024.
27. T. Saito, and M. Rehmsmeier, *The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets*, PLOS ONE, vol. 10, no. 3, p. e0118432, 2015.
28. K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, *The balanced accuracy and its posterior distribution*, in Proceedings of the 2010 20th International Conference on Pattern Recognition, pp. 3121–3124, 2010.
29. T. Fawcett, *An introduction to ROC analysis*, Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.