

Article

Detecting AI-Generated Network Traffic Using Transformer–MLP Ensemble

Byeongchan Kim , Abhishek Chaudhary and Sunoh Choi *

Department of Software Engineering, Jeonbuk National University, Jeonju 54896, Republic of Korea; bcad460@gmail.com (B.K.); abhishek.chaudhary@jbnu.ac.kr (A.C.)

* Correspondence: suno7@jbnu.ac.kr

Abstract

The rapid growth of generative artificial intelligence (AI) has enabled diverse applications but also introduced new attack techniques. Similar to deepfake media, generative AI can be exploited to create AI-generated traffic that evades existing intrusion detection systems (IDSs). This paper proposes a Dual Detection System to detect such synthetic network traffic in the Message Queuing Telemetry Transport (MQTT) protocol widely used in Internet of Things (IoT) environments. The system operates in two stages: (i) primary filtering with a Long Short-Term Memory (LSTM) model to detect malicious traffic, and (ii) secondary verification with a Transformer–MLP ensemble to identify AI-generated traffic. Experimental results show that the proposed method achieves an average accuracy of $99.1 \pm 0.6\%$ across different traffic types (normal, malicious, and AI-generated), with nearly 100% detection of synthetic traffic. These findings demonstrate that the proposed dual detection system effectively overcomes the limitations of single-model approaches and significantly enhances detection performance.

Keywords: MQTT; IoT; DoS; LSTM; transformer; MLP; ensemble; AI; GAN; GPT



Academic Editor: George Drosatos

Received: 15 September 2025

Revised: 17 October 2025

Accepted: 21 October 2025

Published: 22 October 2025

Citation: Kim, B.; Chaudhary, A.; Choi, S. Detecting AI-Generated Network Traffic Using Transformer–MLP Ensemble. *Appl. Sci.* **2025**, *15*, 11338. <https://doi.org/10.3390/app152111338>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Internet of Things (IoT) has rapidly expanded across diverse sectors such as smart grids, factories, and cities [1–3]. According to Statista, the global IoT market size reached USD 1.18 trillion in 2023 and is projected to grow at a CAGR of 12.6% to USD 2.23 trillion by 2028 [4]. As this ecosystem evolves, the Message Queuing Telemetry Transport (MQTT) [5] protocol has become a core communication standard because of its lightweight, publish–subscribe architecture and low-latency characteristics.

However, MQTT still suffers from critical security vulnerabilities. Reports by the Korea Internet & Security Agency (KISA) and the Ministry of Science and ICT (MSIT) show that IoT-related attacks are increasing annually [6]. In 2021, open-source MQTT brokers such as VerneMQ, Mosquitto, and EMQX were found to contain Denial-of-Service (DoS) vulnerabilities (CVE-2021-22116, CVE-2021-33175, CVE-2021-33176) [7]. Furthermore, over 32,000 publicly exposed MQTT brokers were identified without proper authentication mechanisms, potentially allowing attackers to gain remote control over connected devices [8].

As IoT networks grow in complexity, adversaries increasingly use generative AI to craft traffic that closely mimics normal device behaviour. Traditional LSTM-based intrusion detection systems (IDSs) show strong performance against conventional malicious traffic but struggle to detect such synthetic network patterns. Recent studies, including

Ullah et al. (2024) [9], have explored Transformer-based IDS frameworks that leverage transfer learning to enhance detection robustness in AI-driven cybersecurity, highlighting the shift toward generative-AI-resilient architectures.

Building on these insights, this study proposes a Dual Detection System capable of accurately identifying normal, malicious, and AI-generated traffic in MQTT environments. The proposed system first applies LSTM-based primary filtering to distinguish normal from malicious traffic, then employs a Transformer–MLP ensemble in a secondary verification stage to detect synthetic traffic through message-type transition analysis.

The main contributions of this paper are summarised as follows:

- We propose a dual-stage intrusion detection system that combines a lightweight LSTM-based primary detector for conventional malicious traffic with a Transformer–MLP ensemble secondary verifier designed to identify AI-generated traffic.
- We introduce a message-type transition–based feature representation tailored to the MQTT protocol, enabling the detection of adversarial traffic that imitates legitimate communication patterns but fails to reproduce deeper protocol-level behaviours.
- We present a log-ratio–based anomaly scoring mechanism that captures subtle probabilistic irregularities while reducing false positives, thereby enhancing overall detection reliability.
- We validate the proposed approach through extensive experiments on the MQTTset dataset under two representative scenarios (DoS and multi-attack), achieving an average accuracy of 99.1% and 100% detection of AI-generated traffic. Additionally, we analyse inference latency, computational complexity, and trade-offs between detection accuracy and real-time deployment feasibility to provide practical insights for real-world IoT applications.

The remainder of this paper is organised as follows. Section 2 surveys malicious network traffic detection and recent work on generative-AI–based data generation and detection. Section 3 details the proposed dual detection system: (i) an overview of the MQTT protocol; (ii) preprocessing and LSTM-based detection of malicious traffic; (iii) generation of synthetic traffic using GAN and GPT-4o; and (iv) a Transformer–MLP ensemble that detects AI-generated traffic via message-type transitions. Section 4 presents experiments and results under DoS and multi-attack scenarios. Section 5 discusses implications, limitations, and deployment considerations. Section 6 concludes and outlines future research directions.

2. Related Work

This section reviews prior studies relevant to intrusion detection in IoT environments, with a particular focus on approaches addressing malicious network traffic, the rise of AI-generated synthetic data, and detection strategies. It also identifies existing research gaps that motivate the development of the proposed dual detection framework. It is divided into two parts: (1) Malicious Network Traffic Detection, (2) Generative AI-Based Data Generation and Detection.

2.1. Malicious Network Traffic Detection

Research on detecting abnormal traffic in IoT Environments has gained significant attention in recent years. Early approaches employed traditional machine learning and ensemble techniques, achieving high accuracy in identifying common attack types. For instance, Bazaluk et al. (2024) proposed a pre-trained Transformer model and confirmed that pre-training on large-scale MQTT traffic enables robust classification performance even in small-data scenarios [10]. Similarly, Al Hanif and Ilyas (2024) combined the top 10 features selected through feature engineering with ensemble methods such as Random

Forest and XGBoost, enhancing the detection accuracy of DoS Attacks and Brute-Force Attacks to over 95% [11].

More recent studies have focused on MQTT-specific traffic characteristics. Choi and Cho (2022) [12] introduced a Seq2Seq-based malicious MQTT traffic Detection method using the MQTTset Dataset [13]. They extracted five protocol-specific features: Source Port Index, TCP Length, Keep Alive, and Connection ACK, and demonstrated that these features, when input into a sequence model, enabled accurate classification of DoS, SlowITe, Flood, Malformed, and Brute-Force Attacks [12]. Building on their preprocessing methodology, our study employs the same protocol-specific feature extraction strategy to ensure consistent data representation and fair comparison in subsequent experiments.

2.2. Generative AI-Based Data Generation and Detection

In parallel, Generative AI has emerged as both a powerful tool and a new source of threats in network security. Goodfellow et al. (2014) demonstrated that Generative Adversarial Networks (GANs) [14] generate synthetic traffic through adversarial training, while Radford et al. (2018) showed that Generative Pre-trained Transformers (GPTs) [15] generate contextually consistent data using large-scale language models.

Lee and Kim (2024) further demonstrated that GAN and GPT models can generate synthetic traffic nearly indistinguishable from normal traffic in MQTT environments, effectively bypassing existing IDSs [16]. Their findings highlighted a critical gap: while conventional IDSs excel at detecting malicious traffic, they are insufficient for defending against AI-generated traffic.

Recent efforts in AI-generated content detection, such as DetectGPT [17] and GPTZero [18], illustrate how distribution-level characteristics, including perplexity and burstiness, can differentiate synthetic from human-generated data. Inspired by this principle, our study extends distributional analysis to the network domain by applying message-type transition modelling to identify AI-generated MQTT traffic.

Recent advances have explored multiple strategies to enhance IoT intrusion detection. To address privacy concerns, Elaziz et al. (2025) proposed a federated learning framework that decentralizes model training across distributed devices, thereby mitigating data exposure risks [19]. Others have focused on countering adversarial evasion attacks. Yuan et al. (2024), for instance, introduced a hybrid framework that first detects adversarial examples; if an input is flagged, it is rerouted to a conventional machine learning model for more robust classification [20]. Taking a different approach to transparency and robustness, Wali et al. (2025) employed explainable AI (XAI) to verify the credibility of predictions after they are made, re-evaluating suspicious outputs to filter malicious traffic [21].

While these approaches strengthen IDS through privacy-preserving training, hybrid classification strategies, or post hoc explanation, they primarily focus on model-level enhancements against conventional attacks or adversarial perturbations. However, despite these advancements, most approaches still fail to address the distinct challenge of synthetic network traffic generated by advanced generative models, which can closely mimic benign MQTT behaviour and bypass traditional detection methods.

Our approach specifically targets this research gap by shifting from feature-level classification to distribution-level modelling. Drawing inspiration from language modelling principles, we treat MQTT message types as sequential “tokens” and their transitions as contextual dependencies. This approach enables the detection of subtle distributional deviations in synthetic traffic, irregularities that conventional IDS techniques often fail to capture.

In summary, while previous studies have advanced intrusion detection through feature engineering, ensemble learning, and adversarial robustness, they still fall short in

addressing the detection of AI-generated traffic that closely mimics legitimate MQTT communication patterns. Our work addresses this challenge by introducing a protocol-aware, distribution-based detection paradigm that significantly enhances detection accuracy against synthetic traffic.

3. Fake Network Traffic Detection

This section describes the detailed design of the proposed Dual Detection System. First, Section 3.1 explains the structure and characteristics of the MQTT Protocol. Second, Section 3.2 introduces the method of Malicious Network Traffic Detection using the LSTM model. Third, Section 3.3 describes the generation of Fake Traffic through Generative AI to evade existing detection systems. Fourth, Section 3.4 presents the Transformer–MLP ensemble model that leverages message-type transition patterns for secondary verification.

As shown in Figure 1, the flow diagram of the proposed Dual Detection System consists of two sequential steps. In the first stage, the LSTM-based detector effectively identifies conventional malicious traffic but struggles with AI-generated synthetic traffic. To address this limitation, the second stage employs a Transformer–MLP ensemble that re-examines traffic classified as benign, enabling precise detection of synthetic traffic that mimics normal communication patterns.

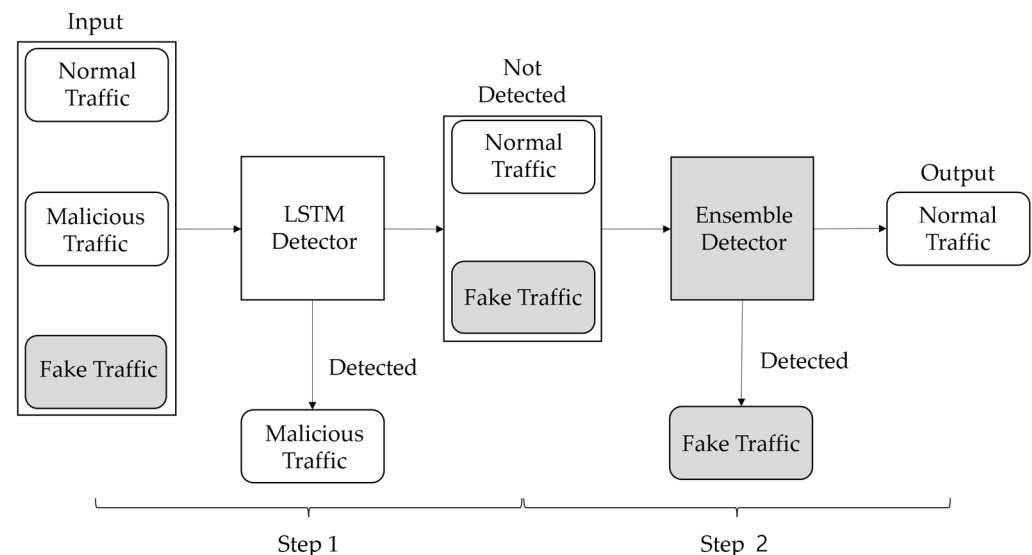


Figure 1. Flow Diagram of the Proposed Dual Detection System.

3.1. MQTT

This subsection provides an overview of the MQTT protocol, which underpins IoT communication. Understanding its structure and components is essential before designing effective intrusion detection mechanisms. MQTT is a lightweight publish/subscribe messaging protocol widely adopted in IoT systems. It consists of three core entities: the publisher, which generates and sends data; the broker, which relays messages; and the subscriber, which receives data based on subscribed topics. As illustrated in Figure 2, publishers and subscribers do not communicate directly but exchange messages only through the broker, ensuring scalability and low network overhead. This message-oriented architecture is particularly relevant to intrusion detection, as abnormal patterns in topic distribution, message-type, or connection control can reveal malicious or AI-generated network behaviour.

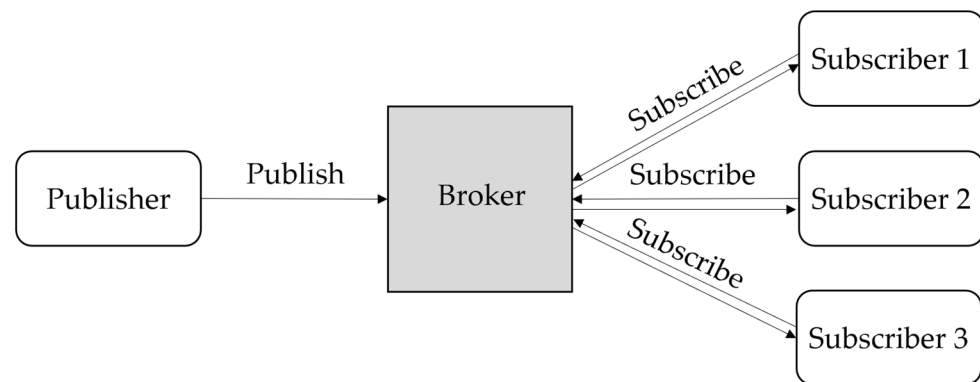


Figure 2. MQTT system structure.

3.2. Malicious Network Traffic Detection (LSTM)

Building on the previous subsection, which introduced the fundamental structure of the MQTT protocol and its relevance to intrusion detection, this subsection presents the primary detection stage of the proposed system. Specifically, we employ an LSTM model trained on protocol-specific features to effectively distinguish normal traffic from malicious MQTT traffic.

3.2.1. Dataset Preprocessing and Feature Extraction

In this study, the MQTTset Dataset released by Vaccari et al. [13] was used, and five protocol-specific features (Source Port Index, TCP Length, MQTT Message Type, Keep Alive, and Connection ACK) were extracted following the preprocessing method proposed by S. Choi [12]. These features serve as the complete input set for the LSTM-based primary detection model. The preprocessing procedure and the extracted features are as follows:

- **Source Port Index**
To reflect the characteristic behaviour of a single publisher initiating multiple ephemeral connections during DoS or Brute-force attacks, each new source port was assigned a sequential index. For example, the first port was indexed as 1001, and subsequent ports were incremented by 1. Frequent port changes indicate a high connection attempt rate typical of resource-exhaustion or brute-force login attempts.
- **TCP Length**
In Flood attacks, attackers typically transmit unusually large packets to overload the broker. To capture this behaviour, packets with payload sizes below 10,000 bytes were normalised to -1 , while packets exceeding this threshold retained their original length values.
- **MQTT Message Type**
The frequency and sequence of MQTT message-types can vary significantly during Flood and Malformed attacks. For instance, flood attacks generate a high volume of PUBLISH packets (type 3), while malformed attacks exhibit abnormal sequences such as a SUBSCRIBE request immediately followed by a PUBLISH message, which attempts to trigger exceptions on the broker.
- **Keep Alive**
In SlowITe attacks [22], attackers exploit the broker's timeout mechanism by sending connection packets with abnormally high Keep Alive values (e.g., 65,535), forcing the broker to wait excessively before terminating the session. Values below 1000 were normalised to -1 , while higher values were preserved to capture this anomaly.
- **Connection ACK**
In Brute-force attacks, repeated authentication failures result in frequent CONNACK

responses with refusal codes (e.g., 5). These raw response codes were retained as they strongly indicate repeated failed login attempts.

These five features are closely associated with distinct attack types, as summarized in Table 1. For instance, repeated source port changes indicate DoS and brute-force activity, while large packet sizes and abnormal message sequences are signatures of flood or malformed traffic. Abnormally high keep-alive intervals and repeated CONNACK errors correspond to SlowITe and brute-force attacks, respectively.

Table 1. Relation between MQTT attacks and features. Note that the circle in the table means that the feature is used to detect the attack.

	Source Port Index	TCP Length	Message Type	Keep Alive	Connection ACK
DoS Attack	○	-	-	-	-
Flood Attack	-	○	○	-	-
SlowITe Attack	-	-	-	○	-
Malformed Attack	-	-	○	-	-
Brute-force Attack	○	-	-	-	○

To detect MQTT traffic, each packet $p_{i,j}$ was represented as a five-dimensional vector of the extracted features, as expressed in Equation (1):

$$p_{i,j} = \{source\ port\ index, TCP\ length, message\ type, keep\ alive, connection\ ACK\} \quad (1)$$

Each client c_i transmits multiple packets, as expressed in Equation (2).

$$c_i = \{p_{i,1}, p_{i,2}, \dots, p_{i,n}\} \quad (2)$$

For model input, k consecutive packets were grouped into a sequence $s_{i,j}$, as expressed in Equation (3):

$$s_{i,j} = \{p_{i,j}, p_{i,j+1}, \dots, p_{i,j+k-1}\} \quad (3)$$

In this study, each packet was converted into a five-dimensional vector, and 16 consecutive vectors were combined into a single sequence. Thus, one sequence was represented by a total of $16 \times 5 = 80$ input values. Each of the five features was selected because it directly captures anomalies tied to specific attack types: for instance, unusually high Keep Alive values indicate SlowITe Attacks, repeated Source Port Index increments reflect Brute-Force or DoS behaviours, and abnormal message-type transitions reveal distorted communication flows. By combining 16 consecutive packets into a single sequence (80 values), the model preserves both local and sequential attack patterns, which are essential for robust detection.

3.2.2. Malicious Network Detection Model

In the primary filtering stage of this study, an LSTM (Long Short-Term Memory) Model was applied, taking the five preprocessed features as input. The LSTM model, a variant of Recurrent Neural Networks (RNNs), is characterized by memory cells and gating mechanisms that effectively combine long-term and short-term information.

Each MQTT traffic sample consists of 16 groups, with five features per group, resulting in a total of 80 values. The primary filtering by the LSTM model rapidly classifies the overall sequence pattern, thereby efficiently distinguishing between normal traffic and malicious traffic. Owing to its memory cell structure, the LSTM effectively learns temporal dependencies across multiple packets, making it well-suited for recognizing repetitive patterns typical of DoS or Flood Attacks. However, while effective against conventional

malicious traffic, its reliance on learned historical patterns limits its ability to detect fake traffic generated by AI, which mimics normal traffic distributions. To address this limitation, Section 3.4 introduces a specialized feature selection and detection method that focuses on message-type transition patterns.

As shown in Figure 3, the traffic is first preprocessed and then passed through the LSTM model, which determines whether the input is normal or malicious.

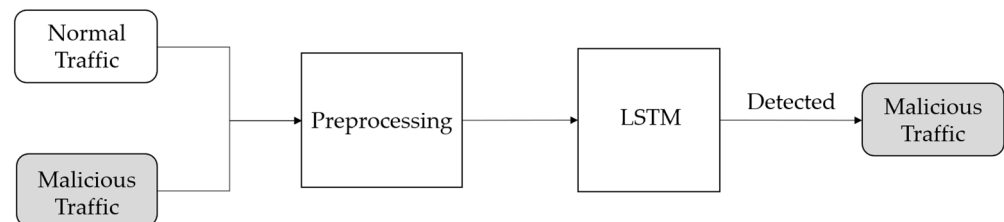


Figure 3. Malicious MQTT Traffic Detection System Structure.

We use a single-layer LSTM (hidden size 128, input size 1) over sequences of length 80 (16 groups $\times$ 5 features). The output of the final time step is passed to a linear layer (128 $\rightarrow$ 1) that outputs logits. We train with BCEWithLogitsLoss and Adam (learning rate 5×10^{-4} , batch size 32) for 30 epochs. During evaluation, we apply a sigmoid to logits to obtain probabilities, and the decision threshold is selected on the validation set per fold (maximizing accuracy) rather than fixed at 0.5.

3.3. Fake Network Traffic Generation

While the previous subsection addressed detection of conventional malicious traffic, here we simulate adversarial attempts to evade IDS by crafting fake traffic that closely mimics normal behaviour. We use a GAN (Generative Adversarial Network) [14] and GPT-4o (Generative Pre-trained Transformer 4 Omni) [23] to generate realistic sequences and then mix the generated and malicious traffic at a 1:15 ratio to build test datasets for evaluation.

As shown in Figure 4, the GAN generator maps random noise to realistic message sequences, while the discriminator learns to distinguish fake from normal traffic. This adversarial training progressively aligns the generated distribution with normal traffic, after which the generated traffic is mixed with malicious traffic to form fake traffic.

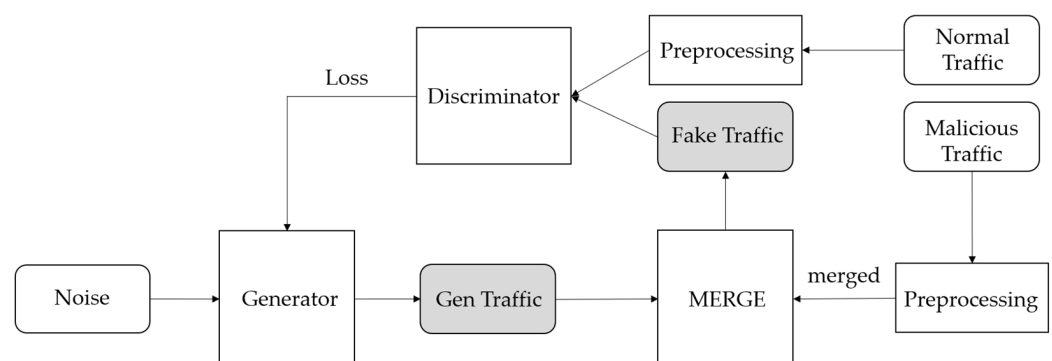


Figure 4. GAN-based Fake MQTT Traffic Generation Architecture.

In addition, we use GPT-4o to generate text-based fake traffic via a prompt-driven strategy: preprocessed normal sequences serve as conditioning inputs, allowing the model to generate non-redundant samples with identical length and structure that closely mimic real MQTT traffic patterns. These GPT-generated samples are then combined with malicious traffic to construct fake traffic. Figure 5 shows the GPT-based generation pipeline; the full prompt template is provided in Appendix A.

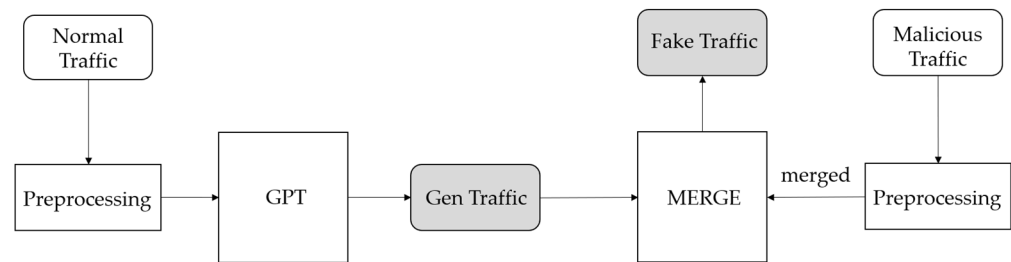


Figure 5. GPT-based Fake MQTT Traffic Generation Architecture.

For the GAN, the generator maps a 100-dimensional noise vector to an 80-dimensional sequence using three linear blocks: $100 \rightarrow 1024 \rightarrow 512 \rightarrow 80$ with BatchNorm (except the last layer), LeakyReLU(0.2), and Tanh at the output. The discriminator uses $80 \rightarrow 512 \rightarrow 256 \rightarrow 1$ with Dropout(0.3), BatchNorm at the second block, LeakyReLU(0.2), and a sigmoid output.

We train both with BCE loss and Adam (learning rate 5×10^{-4} , betas = (0.5, 0.9), batch size 128) for 50 epochs, and apply label smoothing (targets 0.1/0.9) to stabilize training. Inputs are min-max normalised to $[-1, 1]$ using the empirical range of normal traffic and denormalised when saving samples. We apply label smoothing by setting the discriminator targets to 0.1 for real normal samples and 0.9 for fake (generator) samples; the generator is trained against a target of 0.1 for discriminator outputs on generated samples.

For GPT-4o generation, we prompt the model with preprocessed normal sequences and request non-redundant samples of identical length (80 values per row). The full prompt template is provided in the Appendix A; we only keep samples that pass deduplication and length checks.

3.4. Fake Network Traffic Detection Using Transformer and MLP

After generating synthetic traffic, the next challenge is detecting it. This subsection introduces a Transformer–MLP ensemble that analyzes message-type transition patterns to identify AI-generated traffic that could evade the LSTM-based primary detector.

3.4.1. Feature Selection for Fake Detection

While the LSTM model in Section 3.2 leveraged all five preprocessed features, its performance degraded significantly when detecting AI-generated fake traffic. This limitation arises because fake traffic closely mimics the statistical distributions of normal traffic across most features, making it difficult for LSTM to differentiate between the two.

To address this limitation, this study focuses on the MQTT message-type, which reflects the inherent request–reply nature of the protocol and exhibits consistent transition patterns in legitimate communications as shown in Table 2. In contrast, AI-generated fake traffic often disrupts these transitions, creating subtle but detectable irregularities.

Table 2. MQTT Message-Type Transition Probability.

Message Type	Next Message Type (Most Frequent)	Transition Probability (%)
1 (CONNECT)	2 (CONNACK)	100
2 (CONNACK)	8 (SUBSCRIBE)	100
3 (PUBLISH)	3 (PUBLISH)	99.45
8 (SUBSCRIBE)	9 (SUBACK)	100
9 (SUBACK)	3 (PUBLISH)	100
12 (PINGREQ)	13 (PINGRESP)	100
13 (PINGRESP)	3 (PUBLISH)	100

Therefore, only the message-type values were extracted and used as inputs for the Transformer–MLP ensemble. This selective focus enables the model to exploit protocol-level sequential dependencies and capture subtle inconsistencies in transition patterns that are often overlooked when all features are used together.

3.4.2. Detection Model

As shown in Figure 6, traffic initially classified as benign by the LSTM-based primary filter is re-examined by a Transformer–MLP ensemble. In this secondary verification stage, the Transformer models global dependencies across the sequence, whereas the MLP focuses on local token-to-token transitions. From the next-token probability distributions produced by both models, per-step anomaly scores are computed and then combined via a weighted aggregation. This complementary design enables robust detection even when synthetic traffic closely mimics normal communication statistics.

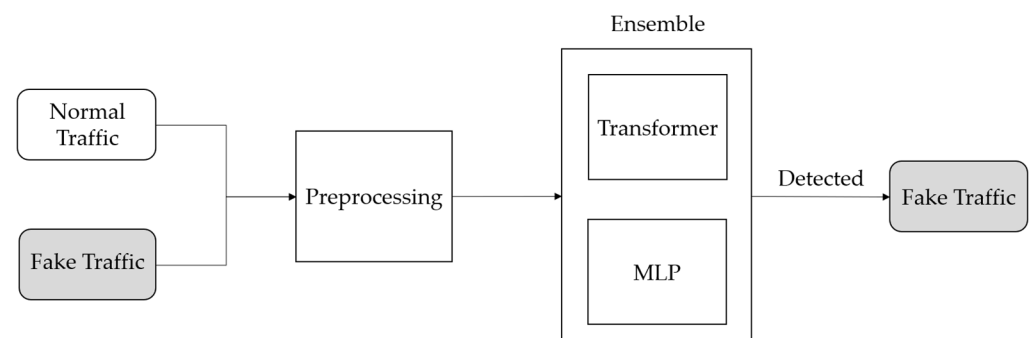


Figure 6. Fake MQTT Traffic Detection System Structure.

Before predicting whether the next message-type follows a normal communication pattern, the Transformer model first analyzes the sequence of previously observed message-types to understand their contextual dependencies. In Equation (4), the transformer takes the sequence of message-type patterns up to the current time step $[x_1, \dots, x_i]$ as input and computes the contextual representation h_i^T .

$$h_i^T = \text{Transformer}([x_1, \dots, x_i]) \quad (4)$$

This representation is then projected into the vocabulary space to estimate the probability distribution of each possible next token. In Equation (5), P_i^T denotes the probability that the next token is k . Here, W^T and b^T are learnable projection parameters, and T represents the Transformer model.

$$P_i^T = \text{softmax}(W^T h_i + b^T) \quad (5)$$

The MLP embeds the current input into a hidden representation to capture nonlinear relationships. First, in Equation (6), the embedding vector e_i is obtained from the previous token x_i

$$e_i = E[x_i] \quad (6)$$

Next, In Equation (7), the embedding is transformed through a nonlinear hidden layer to obtain the hidden representation h_i^M .

$$h_i^M = \sigma(W_1^M e_i + b_1^M) \quad (7)$$

Finally, In Equation (8), the next-token probability distribution P_i^M is computed via a softmax output layer, and $\sigma(\cdot)$ represents the ReLU activation function. Here, $W_1^M, b_1^M, W_2^M, b_2^M$ are learnable parameters, and M denotes the MLP model.

$$P_i^M = \text{softmax}(W_2^M h_i^M + b_2^M) \quad (8)$$

To evaluate how well each model's prediction matches the ground-truth token, a log-ratio score is calculated. This score compares the probability of the true token with the model's most confident prediction, emphasizing large deviations while reducing the penalty for small differences. In other words, it quantifies how confidently the model predicts the correct next token relative to its most likely alternative.

First, in Equation (9), the unit score s_i^T for the Transformer is computed based on the probability of the actual (ground-truth) token $P_{i,true}^T$ and the maximum predicted probability $P_{i,max}^T$.

$$s_i^T = \log \frac{P_{i,max}^T}{P_{i,true}^T} \quad (9)$$

Next, in Equation (10), the unit score s_i^M for the MLP is similarly calculated using $P_{i,true}^M$ and $P_{i,max}^M$. These per-step scores are then used to evaluate the consistency of observed message sequences with normal communication patterns.

$$s_i^M = \log \frac{P_{i,max}^M}{P_{i,true}^M} \quad (10)$$

To capture the complementary strengths of both models, the Transformer and MLP scores are combined through a weighted sum. The resulting sequence-level score reflects the overall likelihood of the observed sequence being consistent with normal communication, serving as the final anomaly indicator.

If only the actual probability $P_{i,true}$ were used as the score, small differences from the maximum probability $P_{i,max}$ could result in excessively large penalties. To mitigate this, the present study computes the relative difference between the actual probability and the maximum probability as a logarithmic ratio, thereby attenuating small differences while emphasizing larger ones.

As expressed in Equation (7), the time-step scores from the two models are combined through a weighted sum to obtain the sequence-level score S_i . The overall sequence score S is then calculated as the average of all time-step scores. Here, L denotes the sequence length, and the weights w_1 and w_2 are parameters that control the relative contributions of the Transformer and the MLP model, respectively.

$$S_i = \frac{w_1 s_i^T + w_2 s_i^M}{w_1 + w_2} \rightarrow S = \frac{1}{L} \sum_{i=1}^L S_i \quad (11)$$

Finally, the averaged sequence score is compared against a threshold. Sequences with scores above the threshold are classified as synthetic traffic, while those below are considered normal traffic.

As expressed in Equation (8), when the overall sequence score S exceeds the threshold τ , the sequence is classified as AI-generated (fake traffic); otherwise, it is classified as normal traffic.

$$\hat{y} = \begin{cases} 1 \text{ (Fake)}, & S > \tau \\ 0 \text{ (Normal)}, & S \leq \tau \end{cases} \quad (12)$$

The threshold τ serves as the criterion for distinguishing between normal traffic and fake traffic based on the final score S . The parameters w_1 , w_2 and τ were all selected according to the F1-score. This hyperparameter search was conducted only on the training and validation datasets, while the final evaluation was performed on an independent test set. This ensures that the thresholding process remains unbiased and generalisable to unseen data.

Only the MQTT message-type tokens are used (range 0–15; out-of-range mapped to 16). The Transformer predictor uses embedding size 128, 2 encoder layers, and 4 attention heads

with sinusoidal positional encoding over length-16 sequences; it is trained to predict the next token at each time step with cross-entropy and Adam (learning rate 1×10^{-3} , batch size 16) for 10 epochs.

The one-step MLP takes the previous token and outputs a next-token distribution: an embedding layer (dim 64) $\rightarrow$ Linear $\rightarrow$ ReLU $\rightarrow$ Linear $\rightarrow$ softmax. It is trained with Adam (learning rate 1×10^{-3} , batch size 32) for 10 epochs using a KL-divergence objective to match the empirical transition distribution (temperature = 2.0).

At inference, we compute per-step log-ratio scores from both models and aggregate them with weights (w_1 for the Transformer and w_2 for the MLP). The weights (w_1 , w_2) and the decision threshold τ are optimised on the validation set per scenario based on the F1-score. The representative results of this optimisation are as follows: DoS (Ensemble): $w_1 = 0.5$, $w_2 = 0.5$, $\tau = 0.7323$; DoS (Transformer-only): $w_1 = 1$, $w_2 = 0$, $\tau = 0.9724$; DoS (MLP-only): $w_1 = 0$, $w_2 = 1$, $\tau = 0.6573$; All types (Transformer-only/Ensemble): $w_1 = 1$, $w_2 = 0$, $\tau = 1.0009$; All types (MLP-only): $w_1 = 0$, $w_2 = 1$, $\tau = 0.0634$. These values represent the best-performing settings discovered via validation, rather than fixed constants.

To prevent overfitting, 5-fold cross-validation was employed, and model selection was based on validation performance consistency across folds. The detailed hyperparameters and optimisation settings for the Transformer and MLP are described in Section 4.1.

3.5. Dual Detection Workflow Summary

Finally, to summarize the overall process, this subsection consolidates the two detection stages into a single sequential workflow, highlighting how they complement each other to provide a robust defence against both conventional and AI-generated threats.

The complete Dual Detection System operates sequentially in two stages. In Step 1, the LSTM model acts as a primary filter, effectively classifying normal and malicious traffic while passing traffic classified as normal to the next stage. In Step 2, the Transformer–MLP ensemble re-examines this traffic to determine whether it is normal or AI-generated fake traffic by analyzing message-type transition patterns. This workflow ensures that conventional malicious traffic is efficiently filtered in the first stage, while sophisticated AI-generated traffic that closely mimics normal behaviour is reliably identified in the second stage.

4. Experimental Results

This section evaluates the performance of the proposed Dual Detection System through four stages of experiments. Section 4.1 describes the dataset composition, experimental environment, and evaluation metrics. Section 4.2 evaluates the detection performance of the LSTM model for normal traffic and malicious traffic. Section 4.3 examines the limitations of the LSTM model in detecting fake traffic generated by Generative AI. Section 4.4 verifies the detection performance of the Transformer–MLP ensemble in identifying fake traffic generated by Generative AI. Finally, Section 4.5 presents the performance of the Dual Detection System, which integrates the primary LSTM model and the secondary Ensemble model.

4.1. Experiment Setup

In this study, two experimental scenarios were defined. The first experiment focused on detecting DoS Attacks, while the second experiment targeted the detection of five types of attacks: DoS Attack, Flood Attack, SlowITe Attack, Malformed Attack, and Brute-Force Attack.

For the first scenario, i.e., the DoS Attack detection experiment, the dataset composition is summarized in Table 3. The training data consisted of 40,000 samples each of normal traffic and malicious traffic (DoS). The test data included 10,000 samples each of normal traffic, malicious traffic (DoS), fake traffic (GPT-4o), and fake traffic (GAN).

Table 3. Dataset Composition for DoS Scenario.

Data Type	Train Data	Test Data
Normal	40,000	10,000
Malicious (DoS)	40,000	10,000
Fake (GPT-4o)	-	10,000
Fake (GAN)	-	10,000

For the second experiment, referred to as All Types scenario, the dataset composition is summarized in Table 4. The training data consisted of 12,000 samples each of normal traffic and malicious traffic (All types). The test data included 3000 samples each of normal traffic, malicious traffic (All types), fake traffic (GPT-4o), and fake traffic (GAN).

Table 4. Dataset Composition for All Types Scenario.

Data Type	Train Data	Test Data
Normal	12,000	3000
Malicious (All types)	12,000	3000
Fake (GPT-4o)	-	3000
Fake (GAN)	-	3000

Tables 3 and 4 summarize the datasets used across all experiments presented in Sections 4.2–4.5. Unless otherwise stated, these splits are consistently applied throughout all experiments, with any deviations explicitly noted where applicable.

All experiments were conducted on an NVIDIA GeForce RTX 4080 Super GPU (NVIDIA, Santa Clara, CA, USA) with Windows 11 (64-bit), Python 3.8, and PyTorch 2.4.1. We applied 5-fold cross-validation to all experiments. Training data were split into five folds; models were selected based on validation consistency across folds, and each fold's trained model was evaluated on the same fixed test set to ensure an unbiased comparison.

All models used the Adam optimiser with learning rates of 5×10^{-4} (GAN, LSTM) and 1×10^{-3} (Transformer, MLP). The loss functions were binary cross-entropy for the LSTM and GAN, cross-entropy for the Transformer, and KL-divergence for the MLP. Hyperparameter optimisation was performed through grid search over the learning rate (1×10^{-3} , 5×10^{-3} , 1×10^{-4}), batch size (16, 32, 128), and hidden dimension (64, 128). The final configuration of each model is detailed in its corresponding section (Section 3.2 for LSTM, Section 3.3 for GAN, and Section 3.4 for Transformer-MLP).

Additionally, we measured the training time for each model. In the DoS scenario, training took 133.60 s for the Transformer model and 253.28 s for the MLP model, while the LSTM model required 936.53 s. In the All Types scenario, the training times were 36.03 s for the Transformer, 77.67 s for the MLP, and 174.83 s for the LSTM.

4.2. Malicious Network Traffic Detection Results

The LSTM-based primary detection model was trained and evaluated under the two experimental scenarios defined in Section 4.1. In this stage, the model was designed as a binary classifier to distinguish between normal and malicious traffic based on the five protocol-specific features described in Section 3.2.1. Fake traffic generated by GAN

or GPT-4o was not included in the training set and was used only in later experiments (Sections 4.3 and 4.4).

Following the dataset splits summarized in Tables 3 and 4, the first experiment focused on detecting a single type of attack (DoS), while the second experiment extended the task to a composite environment containing five attack types: DoS, Flood, SlowITe, Malformed, and Brute-Force. In both cases, the training data consisted of labeled normal and malicious traffic samples, and 5-fold cross-validation was applied to ensure stable evaluation.

Table 5 summarizes the detection results of the LSTM model. In the DoS scenario, the model achieved 100% accuracy for both normal and malicious traffic, demonstrating its strong capability to capture deterministic attack patterns. In the All Types scenario, the model achieved 99.13% accuracy for normal traffic and 94.83% for malicious traffic. Although slightly lower in the more complex environment, the detection performance remained consistently high.

Table 5. Detection Accuracy of LSTM Model for Malicious Traffic.

Experiment	Traffic Type	Accuracy (%)
DoS	Normal (10,000)	100
	Malicious (10,000)	100
All types	Normal (3000)	99.13
	Malicious (3000)	94.83

These results confirm that the LSTM model is highly effective as a primary detection stage against conventional malicious traffic. However, its strong reliance on historical feature patterns indicates a potential limitation: the model may fail to generalise to AI-generated fake traffic, which is further analyzed in Section 4.3.

4.3. Fake Network Traffic Generation Results

This experiment evaluates how the LSTM model, trained solely on normal and malicious traffic, performs when exposed to AI-generated fake traffic that was not included in the training phase. The goal is to assess the model's robustness against unseen adversarial traffic crafted to mimic normal communication patterns.

Following the dataset splits defined in Tables 3 and 4, the model was trained on labeled normal and malicious traffic and then tested against fake traffic generated by GPT-4o and GAN. All performance evaluations were averaged over 5-fold cross-validation.

Table 6 presents the detection results. In the DoS scenario, the model's accuracy dropped significantly when classifying fake traffic, achieving only 66.09% for GPT-4o-generated traffic and 57.9% for GAN-generated traffic. In the All Types scenario, detection accuracy for GPT-4o traffic was relatively high (98.09%), but performance against GAN-generated traffic deteriorated dramatically to 36.87%.

Table 6. Detection Accuracy of LSTM Model for Fake Traffic.

Experiment	Traffic Type	Accuracy (%)
DoS	Fake (GPT-4o)	66.09
	Fake (GAN)	57.9
All types	Fake (GPT-4o)	98.09
	Fake (GAN)	36.87

These findings highlight a key limitation of LSTM-based intrusion detection: despite its strong performance on conventional attacks, it struggles to detect adversarially generated traffic that imitates normal behaviour. The sharp performance gap between GPT-4o and

GAN traffic suggests that while GPT-generated sequences retain subtle statistical differences from normal traffic, GAN-based traffic more closely replicates distributional patterns, making it harder for the model to detect.

This difference can be attributed to the adversarial training mechanism of GANs, in which a generator and a discriminator are trained in competition. The discriminator's objective is to distinguish generated data from real data, while the generator continuously improves to deceive the discriminator. This adversarial process inherently optimizes the generator to bypass deep learning-based detectors such as LSTM. In contrast, GPT-4o relies solely on next-token prediction without such an adversarial feedback loop, resulting in generated traffic that, while similar, is relatively easier to detect.

4.4. Fake Network Traffic Detection Results

In the previous section, the detection performance of the LSTM model against fake traffic was evaluated, revealing significant performance degradation when exposed to AI-generated data. In this section, we evaluate the detection capabilities of three models: Transformer-only, MLP-only, and the Transformer-MLP ensemble. We then analyze their respective strengths and weaknesses individually. Finally, the results are compared with those of the LSTM model to provide a comprehensive understanding of detection performance against AI-generated traffic.

4.4.1. Model Performance Comparison

Table 7 summarizes the detection accuracy of the three detection approaches (Transformer-only, MLP-only, and the Transformer-MLP ensemble) under both the DoS and All Types scenarios. In the All Types scenario, assigning a higher weight to the Transformer significantly improved performance. Consequently, the final ensemble configuration effectively functioned as a Transformer-only model (see Section 3.4) in the All Types scenario, which is reported as “Transformer (Ensemble)” in the table.

Table 7. Detection accuracy of models against fake traffic.

Scenario	Model	Normal	Malicious	Fake (GPT-4o)	Fake (GAN)
DoS	Transformer	100%	99.41%	100%	100%
	MLP	100%	100%	100%	100%
	Ensemble	100%	100%	100%	100%
All types	Transformer (Ensemble)	100%	87.69%	100%	100%
	MLP	91.63%	25.93%	100%	100%

All three models detected GPT- and GAN-generated traffic with 100% accuracy. However, their performance on normal and malicious traffic differed substantially, an important factor for deployment. In the DoS scenario, both the MLP and Ensemble models achieved perfect accuracy due to the relatively simple traffic patterns, but the ensemble's advantage becomes more evident in more complex multi-attack settings (All Types scenario).

The Transformer-only model maintained the highest detection performance across all traffic types, achieving 100% accuracy for normal traffic and 87.69% for malicious traffic in the All Types scenario. This demonstrates its strong capacity to generalise to complex multi-attack conditions. The MLP-only model achieved competitive fake traffic detection performance with extremely lightweight computation but struggled to classify malicious traffic accurately.

The Ensemble model achieved the most balanced performance among the three approaches. By combining the Transformer’s capability to model sequential dependencies across entire message sequences with the MLP’s ability to capture direct token-level relationships, the ensemble improved classification accuracy for normal and malicious traffic while preserving 100% detection for fake traffic.

4.4.2. Inference Efficiency Analysis

To evaluate the practicality of each detection model in real-world scenarios, we measured inference time using 10,000 traffic samples. Table 8 summarizes the results.

Table 8. Inference time and throughput comparison for fake traffic detection.

Model	Total Time (s)	Per Sequence Latency (ms)	Throughput (seq/s)
MLP	88.74	8.874	112.7
Transformer	209.54	20.954	47.7
Ensemble	238.73	23.873	41.9

The MLP model achieved the fastest inference speed, with a throughput of 112.7 sequences per second, which is more than twice as fast as the Transformer and nearly three times faster than the Ensemble. This high efficiency makes the MLP model particularly valuable in scenarios where only fake traffic detection is required, such as real-time network gateways or edge devices with limited computational resources. However, the Ensemble model, while slower, offers a superior trade-off between accuracy and detection robustness, making it a more suitable choice in security-critical environments where precise classification of normal and malicious traffic is equally important.

4.4.3. Comparison with LSTM

Figure 7 compares the detection performance of the Ensemble model with the LSTM baseline for GPT- and GAN-generated traffic. While the LSTM model exhibited severe degradation when detecting GAN-based traffic, the Ensemble consistently achieved 100% detection accuracy across all scenarios.

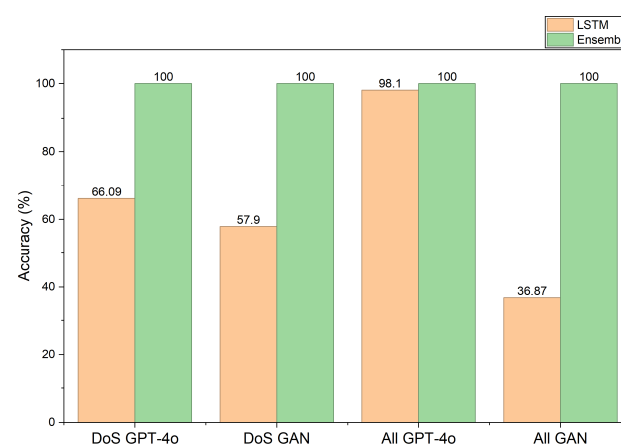


Figure 7. Detection accuracy comparison: LSTM vs. Ensemble.

This direct comparison underscores the fundamental advantage of the proposed Transformer–MLP architecture. By analyzing message-type transition patterns and leveraging complementary detection mechanisms, the ensemble effectively overcomes the limitations of single-model approaches. Furthermore, unlike the LSTM model which struggles

against adversarial distribution matching the ensemble remains consistently robust even when facing sophisticated AI-generated traffic.

4.5. Malicious and Fake Network Traffic Detection Results

Finally, the overall performance of the proposed Dual Detection System, which integrates the LSTM-based primary filtering and the Transformer–MLP ensemble-based secondary verification, was evaluated. In this stage, traffic classified as normal by the first-stage LSTM model was passed to the secondary verification model to determine whether it was AI-generated.

The experiments followed the same dataset configuration defined in Tables 3 and 4, covering both the DoS and All Types scenarios. In the final evaluation, cases where malicious traffic was predicted as fake were also regarded as correct, as the purpose of the second stage is to detect any non-normal traffic patterns, whether they originate from conventional attacks or AI-generated sources.

As summarized in Table 9, the Dual Detection System achieved outstanding detection performance across all traffic types. In the DoS scenario, the system reached 98.9% accuracy for normal traffic and 100% accuracy for malicious and fake traffic (GPT-4o and GAN). In the All Types scenario, detection accuracy was 99.13% for normal traffic and 94.83% for malicious traffic, while fake traffic was detected with perfect accuracy in all cases.

Table 9. Overall Detection Accuracy of Dual Detection System.

Experiment	Traffic Type	Accuracy (%)
DoS	Normal	98.9
	Malicious	100
	Fake (GPT-4o)	100
	Fake (GAN)	100
All types	Normal	99.13
	Malicious	94.83
	Fake (GPT-4o)	100
	Fake (GAN)	100

To further evaluate the system’s detection capability beyond accuracy, we also measured precision, recall, and F1-score, as shown in Table 10. These metrics confirm that the proposed dual-stage system maintains highly reliable detection performance across all traffic categories.

Table 10. Performance metrics of the Dual Detection System.

Scenario	Accuracy	Precision	Recall	F1-Score
DoS	99.73%	99.63%	100%	99.81%
All types	98.49%	99.71%	98.28%	98.99%

As shown in Figure 8, both false positives (FP) and false negatives (FN) occurred only rarely. This demonstrates that the proposed Dual Detection System can accurately classify normal, malicious, and fake traffic simultaneously. Unlike the single LSTM model, which suffered severe degradation when detecting GAN-generated traffic, the proposed two-stage system successfully detected all traffic categories with high precision and recall.

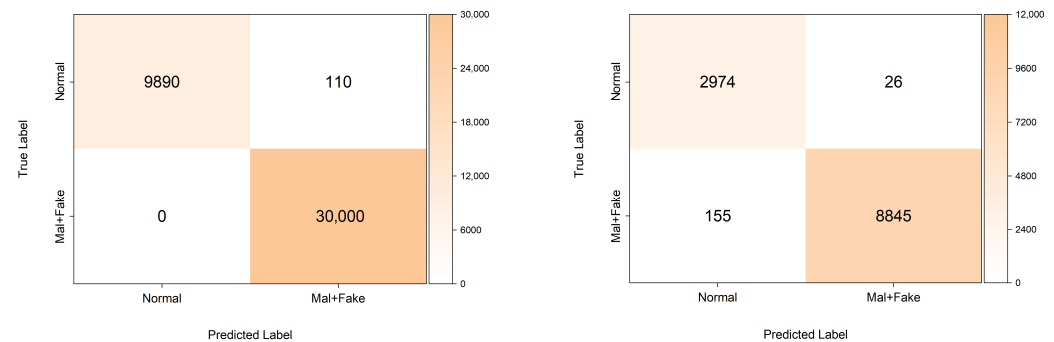


Figure 8. Confusion Matrices of the Dual Detection System.

These results indicate that the Dual Detection System inherits the strengths of both detection approaches: the LSTM provides fast and accurate filtering of conventional malicious traffic, while the ensemble ensures robust detection of AI-generated traffic. On average, the system achieved 99.1% accuracy across all scenarios, confirming its effectiveness and robustness in defending against both traditional and AI-driven threats in MQTT environments.

5. Discussion

This study demonstrated that the proposed Dual Detection System, which integrates an LSTM-based primary detector with a Transformer–MLP ensemble secondary verifier, effectively addresses the critical limitations of existing intrusion detection systems (IDSs) in identifying AI-generated fake traffic. Unlike prior approaches that exhibited significant performance degradation, the proposed framework consistently detected normal, malicious, and fake traffic with high accuracy.

The architecture achieved an average accuracy of 99.1% across all scenarios, confirming its capability to defend against both traditional and emerging threats in real-world IoT deployments. Beyond the numerical results, these findings highlight the importance of message-type transition analysis as a reliable detection feature. MQTT communication follows characteristic message-type sequences; generative models often introduce subtle deviations in these patterns. Leveraging these irregularities enables the system to detect adversarial traffic that conventional IDS methods typically fail to identify.

The complementary strengths of the two components are central to the system's success. The LSTM model provides fast and effective detection of conventional malicious traffic by leveraging temporal feature patterns, while the Transformer–MLP ensemble robustly detects AI-generated traffic by modelling token-level transitions and distributional patterns. As a result, the dual-stage framework achieves stable and resilient performance across a wide range of attack scenarios, demonstrating clear advantages over single-model approaches that struggle with unseen or sophisticated threats.

Nevertheless, the sequential two-stage architecture introduces additional latency, which could affect deployment in real-time or resource-constrained environments. Furthermore, as the experiments were conducted solely on the MQTTset dataset, further validation using diverse traffic sources and real-world network traces is essential to fully assess generalisability.

Finally, while message-type transition analysis is currently highly effective because most generative models fail to replicate deeper protocol-level patterns, future advancements in generative AI may enable adversaries to mimic these message-type sequences more precisely. In such cases, detection based solely on message-type transitions may become less effective, necessitating the integration of additional behavioural or temporal features.

6. Conclusions

This paper presents a Dual Detection System that integrates an LSTM-based primary filtering stage with a Transformer–MLP ensemble secondary verification stage to detect not only normal and malicious traffic but also AI-generated fake traffic in MQTT environments. The proposed framework consistently achieved an average accuracy of 99.1% across all scenarios, with 100% detection of GPT- and GAN-generated traffic. These results demonstrate that the system provides a robust and reliable defence against both conventional and AI-driven threats in IoT networks.

A key contribution of this study is the introduction of message-type transition analysis as a detection feature. By modelling the sequential structure of MQTT communication, the system successfully identifies adversarial traffic that mimics surface-level statistical features but fails to replicate deeper protocol-level behaviour. The addition of a log-ratio-based scoring mechanism further enhances detection reliability by capturing subtle differences in predicted probability distributions, thereby minimising false positives and improving detection robustness.

Despite its strong performance, the current system has certain limitations. The reliance on the MQTTset dataset constrains the scope of evaluation, and future work will aim to validate the framework on more diverse datasets and real-world traffic traces. Additionally, efforts will focus on optimising the architecture for real-time deployment, including lightweight model compression and adaptive thresholding techniques. Expanding this methodology to other messaging protocols such as CoAP and AMQP also represents a promising research direction. Each protocol's unique message structure and communication semantics will require tailored adaptations of the message-type modelling approach.

Overall, this work demonstrates that incorporating protocol-aware message-type modelling with modern deep learning techniques can provide a powerful defence mechanism against evolving generative AI threats. The proposed dual-stage system offers a scalable and extensible foundation for next-generation intrusion detection solutions in real-world IoT deployments.

Author Contributions: Conceptualization, B.K. and S.C.; methodology, B.K.; software, B.K.; validation, B.K.; formal analysis, B.K. and S.C.; investigation, B.K.; writing—original draft preparation, B.K.; writing—review and editing, A.C. and S.C.; supervision, S.C.; project administration, S.C.; funding acquisition, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2023-00237159).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Appendix A

GPT Prompt for Synthetic Traffic Generation

The following is the detailed prompt used for generating synthetic MQTT traffic with GPT-4o.

It ensures that the generated data maintains the original structure while producing non-redundant samples.

Prompt Design:

The following is a preprocessed MQTT Traffic dataset.

Each row consists of 16 groups, and each group contains the following five features: [Source Port Index, TCP Length, MQTT Message Type, Keep Alive, Connection ACK]

Following the above rules, generate data with patterns similar to the original, ensuring that the outputs are not duplicated with the source data and do not duplicate with one another.

The output format should be CSV, and each row must contain 80 numbers (16 groups $\times$ 5 features), separated by spaces.

References

1. Electricity AMI. Available online: <https://www.aitimes.com/news/articleView.html?idxno=141421> (accessed on 22 August 2025).
2. Gas AMI. Available online: <https://www.gasnews.com/news/articleView.html?idxno=104555> (accessed on 22 August 2025).
3. Water AMI. Available online: <https://www.boannews.com/media/view.asp?idx=85538> (accessed on 22 August 2025).
4. Global ICT Research. Available online: https://www.globalict.kr/product/product_list.do?menuCode=040200&knwldNo=143735 (accessed on 22 August 2025).
5. MQTT. Available online: <https://mqtt.org> (accessed on 22 August 2025).
6. IOT Security Threat Report. Available online: <https://www.msit.go.kr/bbs/view.do?sCode=user&nttSeqNo=3185279&pageIndex=&searchTxt=&searchOpt=ALL&bbsSeqNo=94&mId=307&mPid=208> (accessed on 22 August 2025).
7. Three Open Source MQTT Message Brokers Found Vulnerable Against a DoS. Available online: <https://www.secureblink.com/cyber-security-news/three-open-source-mqtt-message-brokers-found-vulnerable-against-a-dos-cyrc-alerted> (accessed on 22 August 2025).
8. 32,000 Smart Homes Can Be Easily Hacked Due to Misconfigured MQTT Servers. Available online: <https://www.csoonline.com/article/566079/32000-smart-homes-can-be-easily-hacked-due-to-misconfigured-mqtt-servers.html> (accessed on 22 August 2025).
9. Ullah, F.; Ullah, S.; Srivastava, G.; Lin, J.C.W. IDS-INT: Intrusion Detection System Using Transformer-Based Transfer Learning for Imbalanced Network Traffic. *Digit. Commun. Netw.* **2024**, *10*, 190–204. [CrossRef]
10. Bazaluk, B.; Hamdan, M.; Ghaleb, M.; Gismalla, M.S.M.; Correa da Silva, F.S.; Batista, D.M. Towards a Transformer-Based Pre-trained Model for IoT Traffic Classification. In Proceedings of the 2024 IEEE Network Operations and Management Symposium (NOMS), Seoul, Republic of Korea, 20–24 May 2024; pp. 1–7.
11. Al Hanif, A.; Ilyas, M. Enhance the Detection of DoS and Brute-Force Attacks within the MQTT Environment through Feature Engineering and Employing an Ensemble Technique. *arXiv* **2024**, arXiv:2408.00480. [CrossRef]
12. Choi, S.; Cho, J. Novel Feature-Extraction Method for Detecting Malicious MQTT Traffic Using Seq2Seq. *Appl. Sci.* **2022**, *12*, 12306. [CrossRef]
13. Vaccari, I.; Chiola, G.; Aiello, M.; Mongelli, M.; Cambiaso, E. MQTTset, a New Dataset for Machine Learning Techniques on MQTT. *Sensors* **2020**, *20*, 6578. [CrossRef] [PubMed]
14. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2690.
15. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. Available online: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 22 August 2025).
16. Lee, S.; Kim, B.; Choi, S. MQTT-based IDS Evasion Method using GPT. *J. Korean Inst. Inf. Technol.* **2024**, *22*, 175–182. [CrossRef]
17. Mitchell, E.; Lee, Y.; Khazatsky, A.; Manning, C.D.; Finn, C. DetectGPT: Zero-Shot Machine-Generated Text Detection Using Probability Curvature. In Proceedings of the 40th International Conference on Machine Learning (ICML), Honolulu, HI, USA, 23–29 July 2023; pp. 24950–24968.
18. Tian, E. GPTZero. Available online: <https://gptzero.me> (accessed on 22 August 2025).
19. Elaziz, M.A.; Fares, I.A.; Dahou, A.; Shrahili, M. Federated learning framework for IoT intrusion detection using tab transformer and nature-inspired hyperparameter optimization. *Front. Big Data* **2025**, *8*, 1526480.
20. Yuan, X.; Han, S.; Huang, W.; Ye, H.; Kong, X.; Zhang, F. A Simple Framework to Enhance the Adversarial Robustness of Deep Learning-based Intrusion Detection System. *Comput. Secur.* **2024**, *137*, 103644. [CrossRef]
21. Wali, S.; Farrukh, Y.A.; Khan, I. Explainable AI and Random Forest Based Reliable Intrusion Detection System. *Comput. Secur.* **2025**, *157*, 104542. [CrossRef]

22. Vaccari, I.; Aiello, M.; Cambiaso, E. SlowITe, a Novel Denial of Service Attack Affecting MQTT. *Sensors* **2020**, *20*, 2932. [[CrossRef](#)] [[PubMed](#)]
23. OpenAI. GPT-4o: Omni Model for Text, Audio, and Vision. Available online: <https://openai.com/index/hello-gpt-4o> (accessed on 22 August 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.