Journal of Cybersecurity and Information Management

♦ ISSN: 2690-6775 Vol: 18, No. 01, 2026 ♦ PP. 01–13

Article DOI: https://doi.org/10.54216/JCIM.180101

FULL LENGTH ARTICLE

A Novel Intrusion Detection Framework Combining Light Feature Engineering, GAN-Based Feature Generation, and Attention-Driven Deep Learning for IoT MQTT Security

Ahmed Dib[1,*] **Zina Oudina**[2] **Sabri Ghazi**[3]

[1]Networks and Systems Laboratory, Badji Mokhtar Annaba University Annaba, Algeria

[2]Embedded Systems Laboratory, Badji Mokhtar Annaba University Annaba, Algeria

[3]Laboratoire de Gestion Electronique de Document – LabGED, Badji Mokhtar Annaba University Annaba, Algeria

Emails: ahmed.dib@univ-annaba.dz; zina.oudina@univ-annaba.org; Sabri.ghazi@univ-annaba.dz

Received: January 03, 2026 Revised: February 05, 2026 Accepted: March 11, 2026 ⋆ Corresponding author

ABSTRACT

MQTT-based Internet of Things networks face major security problems because they have high-dimensional data, class imbalance, and no detection mechanisms that can be understood. This paper proposes a unified intrusion detection framework that integrates attention-based deep learning, GAN-driven data augmentation, and MDA-based feature selection (CNN-LSTM-Attention). The proposed pipeline outperforms both classical and recent state-of-the-art baselines. When tested on MQTTEEB-D, a real-world MQTT dataset with 200,000 flows, an accuracy of 99.12% and macro F1-score of 98.37 were achieved. The attention maps provide clear explanations for the obtained prediction, and the system performs well even against tough attacks such as SlowITe: 96–98%.

Keywords: IoT security ▪ MQTT protocol ▪ Intrusion detection ▪ Feature engineering ▪ MDA ▪ GANs ▪ Class imbalance ▪ Attention mechanisms ▪ Deep learning ▪ Interpretability

1. INTRODUCTION

1.1 Context and motivation: IoT/MQTT security challenges

The use of Internet of Things (IOT) devices has dramatically increased within the last decade. They are present and integrated in most domains of our daily lives and in our workspace. Also in residential, industrial, and medical environments IOT devices are used on a day-to-day basis, such as wireless sensors, intelligent cameras, automatic door openers, air quality measuring devices and many others. By using standardized IOT -protocols as MQTT (Message Queuing Telemetry Transport) it is possible to automate processes, facilitate procedures and get insights. As malicious actors shift their attention towards IoT devices, it has become increasingly important to understand how these devices can be attacked. As the cyber adversary examines the traffic flowing through an IoT network, he will quickly discover vulnerable IoT devices that lack sufficient password protection, do not implement adequate authentication mechanisms, or are not encrypted at all. Given the explosive growth of IoT networks across many industries, security incidents such as malware, ransomware, and botnet attacks are becoming more frequent and more damaging. Each year, the average cost per organization for a security breach has been increasing. Today's enterprise security arc hitecture is no longer sufficient to protect against today's expanding threats. In IoT scenarios, MQTT brokers serve the purpose of providing an interface between IoT devices. They are used to store and send messages from connected devices (publish) to oth er connected devices (subscribe) in order to achieve a messag-

ing hub. For securing IoT networks, it is not only important to prevent unauthorized access and data injection into the network. Rather, there needs to be dealt with the complexity of the heterog eneous IoT environment. Even though it is possible to find uniform protections for various IoT devices and environments, there is considerable variability in the various types of IoT hardware and software. This variety complicates the design of protections . Moreover, scalability becomes a serious problem when a large number of devices are connected to the network. In such networks, thousands of sensors are available, collecting data in real time. Conventional IDS solutions struggle to cope with the large nu mber and variety of possible network traffic within such scenarios, often reaching their limits. As AI technologies like deep fakes, automated scans, and generative social engineering continue to evolve, they have introduced new challenges for defenders who are now forced to leverage adaptive AI-based approaches that learn in order to address these threats. In reality, there is an imbalanced amount of samples for attacks and normal traffic, which makes a great challenge for a detector to learn and distinguish between normal behavior and suspicious behavior. Most existing approaches, such as MQTT-based IDS, heavily rely on manually designed features which may fail to capture slight anomalies that enable attackers to evade detection. Furthermore, state-of-the-art deep learning models are notorious as black- box models, which make it difficult to interpret and make proper decisions based on the learned representations. Due to aforementioned problems, making IoT and MQTT networks more secure is a pressing concern. Solutions should not only be able to guarantee reliability, explainability, efficiency and flexibility, but also leverage advanced techniques such as feature engineering, data improvement and increase in data diversity, and attention mechanisms for protecting computer systems and network infrastructures.

1.2 Limitations of conventional intrusion detection approaches

Static IDS solutions, designed to identify known threats have limitations, especially when based on fixed signatures and not sufficient to handle unknown attacks. Traditional IDS systems struggle to deal with zero -day attacks that evolve daily. Traditional attacks are becoming less effective and are being quickly replaced by more sophisticated methods of infiltration [1]. Highly skilled cybercrimina ls use various techniques for hiding malicious content in normal network packets to bypass static filters, including header modification, steganography and packet reassemble attacks. As data communications are increasingly secured with encryption, traditional signature-based approaches are no longer sufficient. Many conventional models utilize traditional methods to monitor and protect the IoT networks. Some of these systems implement simple machine learning methods to effectively flag up newly found threats, and this approach did have some value when initially rolled out. However, these models relied on basic features and generic statistical techniques. As a result, they are effective at identifying well -known intrusion types; however, more complex, multi-stage, long-term attacks are more difficult to detect [2]. The increased number of IoT devices, each with their own intricacies, presents new challenges to the traditional IDS approach. The same attack can manifest in different ways due to the differing fun ctionality of various devices, and as IoT devices are largely provided by individual companies using custom firmware, it is increasingly difficult for traditional models to generalize and function effectively as the IoT landscape continues to expand. Another challenge relates to scalability. Many existing legacy IDS implementations are optimized for older computing architectures that have sufficient computing power, memory and even dedicated servers to host them. However, as many current legacy systems reac h maturity, scalability becomes a major concern. Threat detection systems have to monitor networks comprised of thousands of IoT devices [3]. However, modern IDS solutions may have memory and computational requirements that are beyond the capability of the endpoint, totem-pole of devices/servers that integrate with the solution, or the local area network infrastructure e.g. a MQTT network. As more devices come online and start to communicate with each other, there is a real risk that traditional IDS may encounter issues such as throughput degradation, packet loss and system overload especially during periods of high traffic volumes or carefully timed attacks. IoT-specific challenges such as time -criticality, data density, diverse data types, multiple sources , and centralization-to-distribution transition complicate the detection process. In addition, most current approaches treat all anomalous events as suspicious, and many of these events are actually not security threats, for example, an event caused by mal function of sensors or transient network fluctuations. In practice, there are many "false positives" in systems that report all anomalous connections, and many of these events are not security threats and only generate a large number of alarms for normal b ehavior [1]. Also, many systems lack alert clarity, such as lacking explanations for why an alert was generated, making it difficult for IoT teams to understand and explain system behavior. Although there are some approaches that have approached the proble m of IoT security using advanced AI -based models for pattern recognition in large datasets, they require large amounts of high -quality labeled attack data, which is hard to obtain for real -world MQTT networks. Also, model update using new data incurs signi ficant computational and memory costs. Moreover, deep learning models are often black boxes and difficult to understand, verify, and explain, representing a significant concern for security and compliance people. Most existing intrusion detection systems t ypically adopt either statistical methods, rule -based methods or ensemble methods in a monolithic way to form hybrid systems, which can improve detection accuracy in certain situations. However, when attacks become more sophisticated by using AI techniques to evade detection or even generate artificial normal data, these methods no longer work effectively. It is challenging to achieve a good balance between the detection efficiency (i.e., inference speed) and the computational efficiency and detection accuracy. Another challenge to detect intrusions on MQTT messages is the MQTT protocol itself. Unfortunately, many of the current commercial IDS systems do not handle some of the MQTT specific features. Handling of sessions, absence of basic security features, and asynchronously sent messages are some of these features. These features require modification to current solutions, adding more complexity, and more maintenance. As time passes, more vulnerabilities are discovered and these solutions require updates to continue to protect resources on your network. The current state of

neural network based network traffic classification models for IoT devices has reached its limitations. Models have to be more adaptive, transparent, interoperable and able to keep up with the dynamic IoT- and MQTT-environment. Currently prevailing "react first and then optimize" approach and the focus on accuracy are no longer sufficient to tackle these new challenges. Smart features, feature generation and improved attention mechanisms have to be integrated into state -of-the-art generative deep learning architectures to push the frontier of IoT security towards next generation models.

1.3 Novelty: combining light feature engineering (MDA), GAN-based feature generation, and attention-driven deep learning

In this paper, a multi -layered approach feature learning, data augmentation and model interpretability for IoT/MQTT security is proposed. Traditional intrusion detection methods have several limitations. Our approach begins with a light feature engineering step. The features are reduced down to the most important ones based on their importances obtained with the Mean Decrease in Accuracy (MDA) method. MDA is an objective feature selection method used to re duce the dimensionality of a dataset by retaining the most relevant features that improve the accuracy of the learning model. For security analysis on IoT data, there is a high likelihood that the number and type of devices escalates rapidly. Moreover, add itional features could add complexity to the model, making it harder for it to learn and deploy on the limited capacity MQTT edge devices. The MDA approach also allows for more freedom in choosing features. While the important variables for an embedded environment typically remain the same, they can differ as IoT scenarios and attacks evolve over time. Our approach to adaptive selection of relevant features can thus always learn from the latest occurrences of unknown attacks. To address the lack of or imbalance in existing real-world IoT data sets, a Generative Adversarial Networks (GAN) is applied to enhance the value of the existing data set. A GAN consists of two neural networks. The two networks are pitted against each other in a competition where one ne twork generates samples (in this case, IoT network traffic) that are real enough that the second network cannot distinguish between them and additional samples that have been drawn from the data set. In practice, GANs are very valuable for cybersecurity wh ere real examples of malicious attacks and targeted cyber -attacks are extremely rare and valuable [6]. In addition to generating more samples, the GAN is particularly useful in simulating real, but rare intrusions and edge-cases, and can even out the class distributions. The third stage of our approach attention -driven deep learning, based on latest achievements in artificial intelligence for language processing, computer vision and analysis of sequential data. Due to low proportion of suspicious events in traffic, the model needs to focus on the most relevant parts of input data, i.e. find attention to needed features [6]. Attention layers allow network to highlight context, which would get lost in the big flow of regular events. In this paper, a novel appr oach is proposed that integrates attention mechanisms into Convolutional Neural Networks (CNN). Using CNNs allows the network to learn spatial relations between features. The attention modules allow the network to track these relations over time [7]. The c ombination of both enables the network to detect a lot of more complex patterns which solves an important part of the security issue:

• MDA (Multivariate Decision Agriculture) very quickly removes non-informative features and identifies those

that are most relevant to the prediction.

• Variety brought to the training ground by GANs, and the resulting fun and games that the network has to play

whilst it is not being attacked and is accumulating learning experiences, in order to overcome issues related to lack of training data and the dynamic threat landscape.

• Building efficient models using high-level attention based deep learning approaches that are aware of context

such as time and location in order to not only able to detect novel sophisticated threats but als o to provide mechanisms to trace back causes of disturbances and enhance system performance and trustworthiness. Most previous works focus on one component, e.g., word embedding, sentence structure, or multi -task deep learning framework. Our approach unifies these perspectives within a single pipeline that improve both accuracy and efficiency, and make the method more transparent.

2. RELATED WORK

2.1 Review of existing feature engineering, GANs, and attention-based models for intrusion detection

The areas of intrusion detection where advanced machine learning meet IoT security is an active area of research. Looking at promising directions from recent literature, three areas emerge: better feature engineering, generative modeling using GANs, and th e application of attention in deep learning models. This post will serve as a high - level survey of related work for a research paper on the topic. In study Nimbalkar and Kshirsagar [8], authors have demonstrated significance of feature selection and experiment with ReliefF and some other feature selection techniques such as Gain Ratio and Information Gain in order to achieve optimal set of features in IoT data streams for improved intelligence in processing large scale IoT datasets by elevating accuracy and greatly reducing computational costs for more intelligent feature selection. Ghubaish et al. in [5] proposed a method called LEMDA (Lightweight Ensemble Mean Decrease in Accuracy). This method auto selects the relevant set of features, using a subset of the feature space that yields highest accuracy levels. The selected feature space changes with emergence of new attack behaviors. Zeghida et al. in [6] use a trained neural network to generate "fake" attack instances to simulate the insertion of new samples in the training set. This adversarial method represents a very important mechanism to balance datasets and to improve the learning capability in detecting less represented attacks. In the related work, Salehiyan et al. [9] presented a GAN-based approach to generating (fake) samples of normal MQTT traffic as well as to learning features that enable a model to not only recognize known traffic signatures but also to guess the signatures of new unknown attack types. Attention-based methods have also started t o emerge in this context. In Yin et al. [10], a hybrid deep learning architecture, built using LSTM and CNN layers with integrated attention is employed for learning effective tem-

poral and spatial features from network data to enhance the capability of an IDS system in detecting potential intrusions. Akuthota and Bhargava [11] proposed an attention based deep learning architecture for IoT. Their Transformer - based IDS could efficiently distinguish unusual communication patterns through huge amounts of device . Alsubaei [12] combined Extreme Gradient Boosting (XGBoost) with deep neural networks for intrusion detection in IoT. They used XGBoost to rank and select the most salient features before putting them into a hybrid deep learning model. This provided more efficient model when making predictions than standalone architectures. Boppana and Bagade [13] also did important work on using Cycle GANs to simulate hard -to-get attack patterns in MQTT datasets that are very unbalanced. They showed that by making synthetic data more varied and realistic, they were able to find rare and advanced persistent threats in IoT networks much better, especially in networks where traditional data augmentation methods don't work. Nadiah et al. [14] utilized the multi-head self-attention mechanism in the widely known Transformer architecture for large-scale IOT intrusion detection. The proposed solution was not only more accurate than conventional rule- based_IDS but also provided an added layer of insight into which device actions and/or sequences were most likely to lead to a breach.

2.2 Comparison with traditional and recent hybrid IDS approaches in IoT

Traditional approaches to IDS for IoT have been mainly based on the combination of existing security solutions with newly developed mac hine learning algorithms. Some innovative solutions also utilize dedicated artificial intelligence. Yet, despite these improvements, there are a number of remaining issues. In [15], Prajisha and Vasudevan proposed a hybrid IDS that combines feature selecti on using LightGBM with DNN. This system produced very accurate results for a wide range of IoT protocols. However, their proposed solution requires significant costs to retrain the model in highly dynamic environments where device behaviors and attack patterns may change rapidly causing feature drift and decreased performance over time. An unsupervised clustering approach that integrates hierarchical neural networks for real -time anomaly detection in IoT networks was recently proposed by Zhu et al. [16]. Although it achieves good results in identifying outlier events in the time domain without relying on manual labeled data, it also generates a large number of unclear anomaly alerts (false positives), which need to be carefully examined and manually corrected, rendering it difficult to scale up to handle large-scale IoT network workloads. In order to monitor MQTT traffic, Siddharthan, et al. in [17] designed an ensemble of learning based Intrusion Detection Systems (IDS) that combined decision tree with LSTM. Although improved detection rates were achieved, increased resource utili zation at times resulted in delayed response particularly at periods of high network utilization. Rahman et al. [18] employed GAN -augmented synthetic data to address class imbalance in industry -specific scenarios for enhancing robustness of IDS for IoT. They were able to successfully identify minorities in data, but adversarial samples remained a challenge that was difficult to counter without re -designing the model pipeline. Alsharaiah et al. [19] designed a Transformer -based attention-based approach called Sharaa in order to identify cross-device anomalies in healthcare IoT. The proposed approach is explainable, and it can even operate on encrypted network traffic. However, a large computational effort has to be exhausted to execute the approaches in existing comp uting environments. Therefore, it is challenging to port the approaches to tiny IoT gateways or battery-powered nodes. In [12], Alsubaei combined XGBoost -based ensemble selection with deep learning and significantly improved the accuracy and efficiency (i.e., reduced inference time) over purely deep models. Nonetheless, as the number of devices increased, the approach failed to scale.

2.3 Summary and Positioning

Table 1 compare existing works along the lines of methodology, data and performance. Unlike previou s works that studied feature selection, GAN augmentation, or attention in isolation, a unified framework is presented that leverages all three techniques and outperforms existing works on the MQTTEEB -D dataset.

Table 1. Comparative Overview of Related Intrusion Detection Works

Reference	Technique	Dataset	Feat. Sel.	GAN	Attn.	Acc.
Nimbalkar & Kshirsagar [8]	ReliefF + Classical ML	IoT datasets (NSL-KDD)	✓	–	–	94.1%
Yin et al. [10]	CNN + LSTM (RNN)	KDD99	–	–	–	93.5%
Siddharthan et al. [17]	Ensemble DT + LSTM	SenMQTT-Set	–	–	–	97.2%
Boppana & Bagade [13]	GAN-AE	MQTT networks	–	✓	–	96.8%
Sasi et al. [24]	1D-CNN-LSTM + Self-Attention	IoT traffic	–	–	✓	98.3%
Proposed Framework	MDA + GAN + CNN-LSTM-Attention	MQTTEEB-D	✓	✓	✓	99.12%

3. DATASET DESCRIPTION

3.1 Overview of the MQTTEEB-D dataset

MQTTEEB-D dataset includes five different attack classes as shown in Table 2 and normal traffic, which represents legitimate behavior including inter-device communication, sensor readings and normal activity behavior patterns.

Table 2. Attack types included in MQTTEEB-D dataset

Attack Type	Frequency	%	Characteristics
DoS	∼35,000	17.5%	High-volume, rapid connection exhaustion
SlowITe	∼28,000	14%	Low-rate, sustained resource consumption
Malformed Data	∼22,000	11%	Packet injection, protocol exploitation
Brute Force	∼25,000	12.5%	Credential enumeration and login attempts
Publish Flooding	∼40,000	20%	Topic-level message spamming
Benign/Normal	∼130,000	25%	Real sensor readings and device communications

MQTTEEB-D: A Real IoT Traffic Dataset. The dataset MQTTEEB-D is an innovative source of information for the testbed of MQTTEEB. It is the record of real IoT traffic from a real implementation of the testbed at the International University of Rabat (UIR) in Morocco. The real world IoT traffic from d ifferent complicated networks has been captured [4].

3.2 Key statistics and attack types included

The MQTTEEB -D dataset provides comprehensive coverage of IoT intrusion detection requirements through multiple dimensions of data.

3.3 Specific challenges: data imbalance, heterogeneous features, real-world scenarios

While MQTTEEB-D is very useful because it is based on real life, its realistic nature also makes it harder to use in practice, just like security professionals have to deal with. Attacks are relativel y rare compared to normal usage in real IoT networks. This is reflected in MQTTEEB -D, which consists mostly of normal traffic (65%) as opposed to attack packets (35%). Furthermore, within the attack packet portion,

different types of attacks are not equall y distributed: for example, Publish Flooding is the most frequent type of attack (20%) whereas Malformed Data is the least frequent (11%).

4. PROPOSED METHODOLOGY

4.1 General Pipeline Overview

The main new thing about this work is that it combines three different AI techniques into a single, end -to-end intrusion detection pipeline as shown in Figure 1.

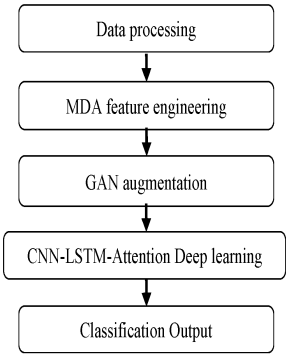

Figure 1. General Pipeline of the proposed solution

A common problem with existing IoT/MQTT IDS is that they treat feature engineering, data augmentation, and deep learning as separate stages. The following synergistic framework is suggested in which each stage improves the next. Figure 2 gives a visual representation of the proposed methodology.

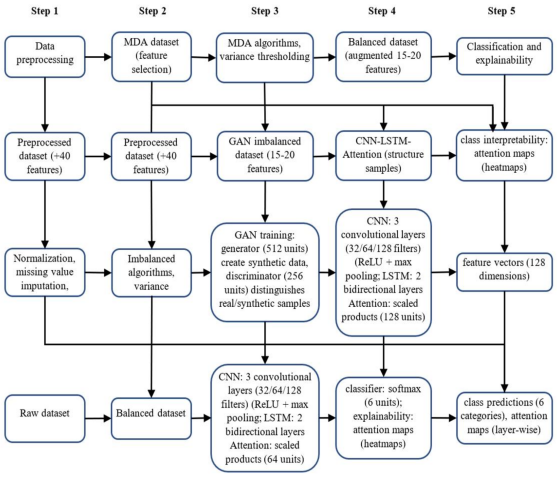

Figure 2. Architectural view of the proposed solution

The pipeline diagram shows how our proposed intrusion detection system works. It is broken down into three main phases.

4.2 Phase 1: Data Preprocessing and Feature Selection

Our initial processing steps begin by cleaning and normalizing the raw data as MQTT ne twork traffic snapshots. A dimensionality reduction is applied through a permutation based feature selection technique (MDA) that effectively decreases our input features from 40+ to 15-20.

• Stage 1: Data Preprocessing

In this step, the data were cleaned, normalized, and key features were identified of the cleaned and normalized MQTT network traffic and then resulting cleaned and feature extracted traffic were used to feed a machine learning engine. ≠ Data raw: Unprocessed MQTT Network Packets from the MQTTEE B-D Dataset. This dataset contains 40 features like Temporal Features (Inter-arrival Times) and Statistical Features describing Packet Size (mean and std-deviation) as well as Protocol Specific Features (e.g., QoS Level, Topic Frequency) and Session Based Features (e.g., Session Duration). ≠ Data preprocessing: This component controls all the necessary data cleaning processes that are performed on the MQTT Network Packets. For the MQTTEEB -D dataset, this component first normalizes the packet data using a min-max scaling between 0 and 1, followed by replacing missing field values using KNN imputation or mean imputation and finally, this component uses statistical methods for Outlier detection and removal. ≠ Preprocessed dataset (40+ features): The output of the workflow is the Preprocessed MQTT Network Packet Dataset.

• Stage 2: MDA Feature Selection

This section describes the main components and steps to perform feature selection. ≠ MDA Dataset (Feature Selection): Main feature selection tool that uses a permutation-based method to determine the importance of features. It analyzes the results to see how predictive each feature is of the class that it is a member of. ≠ The Preprocessed Dataset (baseline) 40+ features: the features provided in Stage 1 with all features included. All the variables from the original dataset have been included in this dataset as the baseline solution. ≠ Algorithms. For better understanding of the data, variance thresholding is used to remove features with low variance that did not contribute to our analysis. Data were reduced from more than 40 dimensions down to 15- 20 very important dimensions. This is the Intermediate Output after removing some features but keeping the class balance. This model has fewer features than the Basic Outp ut, but it is still balanced. It can be used to generate more instances of the smaller feature set in Stage 3.

4.3 Phase 2: GAN-Based Data Augmentation

In order to fulfill the goal of this portion of the work, the term GAN-based data augmentation is utilized to describe the process of generating synthetic samples in the feature -level for the minority class of attacks where GAN is employed to generate synthetic samples of the minority class. For the work presented in this document, GANs are used as a means to alleviate class imbalance whereby SMOTE is used as a preliminary class rebalancing technique in order to stabilize the training of the GAN. Synthetic samples used for training are all generated by the utilized GAN in the feature space. This step of creating synthetic attack samples using GANs in order to alleviate class imbalance and to increase the variety of the training dataset out comprises the following Sequence of Actions: ≠ GAN Imbalanced Data: The input data has the correct features but the data is imba lanced such as 65% benign and 35% attack. ≠ The size of the Generator neural network is set to 512 neurons, and the size of the Discriminator neural network is set to 256 neurons. The Generator is used to generate synthetic samples of attack data, while the Discriminator is used to distinguish between real and synthetic data. This is referred to as adversarial training, where the Generator strives to generate samples that are indistin-

guishable from real samples, and the Discriminator strives to identify synth etic samples and real samples. This section describes how the network processes the additional data that is used in the work presented in this document.

4.4 Phase 3: CNN-LSTM-Attention Classification and Explainability

In this phase, the data is detected and decisions are made using a core detection and decision engine. A balanced and low dimensional feature set is then used to classify the traffic flows and to generate attention maps using a CNN-LSTM-Attention architecture.

• Stage 1: CNN-LSTM-Attention Architecture

In this paper, the term CNN –LSTM–Attention architecture refers to the specific deep learning model composed of convolutional layers for spatial feature extraction, bidirectional LSTM layers for temporal modeling, and an attention mechanism for dynamic feature and time-step weighting. The expression attention-driven deep learning is used solely to describe the underlying design principle, whereby attention mechanisms guide the learning process by emphasizing the most discriminative patterns. For clarity, all experimental results reported in this paper are obtained using the CNN–LSTM–Attention architecture. The objective of this stage is to create a CNN for enhanced extraction and representation of complex spatial and temporal features from the improved dataset. This includes: ≠ Balanced dataset (Augmented Dataset) with a Balanced Class Distribution (15 -20 feature input), an optimum feature set from Stage 3. ≠ The combination of the three modules (CNN, LSTM, Attention) creates a powerful synergy when s ampled together. ≠ Convolutional Neural Networks (CNNs): The CNN portion of this architecture consists of three convolutional (Conv) layers, each containing 32, 64, and 128 filters, respectively. ReLU activations are applied to CNN feature maps and spatial d imension are down -sampled through Max Pooling. An additional Attention component incorporates scaled Dot products of 128 units. ≠ Feature Vector: The Feature Vector represents the 256 -dimensional output representation of the Combined CNN-LSTM-Attention Stack Architecture. For clarity and repeatability, the formal definitions of the LSTM and Attention components in the proposed architecture are specified mathematically. For each time step t, the Bidirectional LSTM computes the following gate operations:

$$i_t = \sigma\left(W_i[h_{t-1}, x_t] + b_i\right), \tag{1}$$

$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right), \tag{2}$$

$$o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right), \tag{3}$$

$$\tilde{C}_t = \tanh\left(W_c[h_{t-1}, x_t] + b_c\right), \tag{4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{5}$$

$$h_t = o_t \odot \tanh(C_t). \tag{6}$$

Here, $x_t \in \mathbb{R}^d$ is the input feature vector at time step t after MDA selection, $h_{t-1} \in \mathbb{R}^u$ and $C_{t-1} \in \mathbb{R}^u$ are the previous hidden and cell states, and i_t, f_t, and o_t are the input, forget, and output gates. The matrices W_i, W_f, W_o, and W_c are learnable weights, while b_i, b_f, b_o, and b_c are bias vectors. The operator $\odot$ denotes element-wise multiplication. For the Bidirectional LSTM, the forward and backward hidden states are concatenated as:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]. \tag{7}$$

The Multi-Head Attention mechanism operates on the LSTM output sequence $H = [h_1, h_2, \ldots, h_T] \in \mathbb{R}^{T \times u}$. For each attention head k ($k = 1, \ldots, K$, with $K = 4$), the scaled dot-product attention is computed as:

$$Q_k = HW_k^Q, \tag{8}$$

$$K_k = HW_k^K, \tag{9}$$

$$V_k = HW_k^V, \tag{10}$$

$$\text{head}_k = \text{softmax}\left(\frac{Q_k K_k^\top}{\sqrt{d_k}}\right) V_k, \tag{11}$$

$$\text{MultiHead}(H) = \text{Concat}(\text{head}_1, \ldots, \text{head}_K) W^O. \tag{12}$$

In these equations, W_k^Q, W_k^K, and W_k^V are projection matrices for query, key, and value representations, $d_k = 64$ is the dimension of each attention head, $\sqrt{d_k}$ stabilizes the softmax computation, and W^O is the output projection matrix. The resulting attention weight matrix A k, referred to here as the attention map whose components allow for interpretation, shows for each time step t, to what degree t attends to any other time step t-. This allows a security operator to not only determine if a time duration of data was found to be classified differently than Normal, but also which individual features contributed to the classification decision.

• Stage 2: Classification and Explainability

While the above stages are intended to assist in the overall goal of network intrusion detecti on, the final stage involves two additional goals: classification of incoming network packet streams into “intrusion detected” or “intrusion not detected” categories, and provision of explanations that security administrators and cyber investigators can understand. In particular, this stage includes a role as a classifier and as an explainability engine, generating human-understandable output from the 128 feature vector values computed in Stage 4. ≠ The output of the Softmax classifier is the Softmax classifi er output on the 6 units that represent one benign class plus 5 different attack types. The output for each class provides a probability score representing the likelihood that the attack is represented in the feature space. ≠ Attention maps or heatmaps gener ated from attention layers of Stage 4 highlight where in each feature set or time intervals each feature contributed the most to the probability score of each predicted class. The heatmaps provide deeper insights into how classes are supported, and reveal key message features such as frequency, entropy, and time irregularities. ≠ Full analysis details such as predicted class and corresponding attention maps will also be available for in - depth analysis. ≠ The combination of prediction and explanation capabilities with confidence scores and practical, interpretable forensic indicators greatly improves the ability to respond to incidents.

4.5 Justification of the architecture choice

The deployment of the combined approach that utilizes feature-selection method of MDA, data augmentation using GAN, and CNN –LSTM–Attention architecture for intrusion detection in IoT networks is supported by current scientific

findings. Each of the employed techniques, by their own merits, enhance the capabilities for detecting intrusions in IoT environments.

• Feature Selection combined with Deep Learning

Combining Feature Selection and Deep Learning results in less error and less cost than using the two methods alone. Feature selection enables the training of a deep learning model to generate less erroneous generalization with lower costs, compared with single use of deep learning. Although a deep learning model trained with an IoT dataset of more than 40 features learns only spurious correlations, the generalization capability of the trained model decreases. However, when using MDA-based selection, the number of selected features is reduced to 15% to 20% of the total number of features. Note that even though the number of selected features is only 15% to 20% of the total features, the features selected using MDA retain approximately 85% to 95% of the prediction accuracy of all features. The balance of speed, strength, and accuracy is therefore maintained by using the two methods [20]. Without using Feature Selection, the model is likely to be overly fitted to the noisy features in the dataset. Without using Deep Learning, the model is likely to miss out on learning the several highly nonlinear interactions that exist between many of the features/variables in the dataset. The combined approach is more robust and provides interpretable models.

• GAN Augmentation combined to Deep Learning

Real-world IoT datasets typically have a 65% to 35% class imbalance in terms of benign versus attack, resulting in high false negative rates when models are developed without data augmentation, meaning they may not capture some attack types that are less common [21]. By producing synthetic attack samples created by GANs that resemble realistic variations (1) to address this class imbalance, (2) create attack types that didn't exist previously in the dataset, and (3) create valid representations of rare attack types.

• Attention Mechanisms combined to CNN-LSTM

CNNs examine the spatial information of patterns, while LSTMs examine the temporal sequences of those patterns; therefore, traditional architectures fail to combine both CNNs and LSTMs to properly specify patterns and detect context-dependent anomalies. By employing attention mechanisms, these architectures can weight the importance of both features and time-frames, which allows them to give more attention to the more relevant signals while reducing noise in the model [7]. Attention-mechanism-based architectures achieve a 5% -15% performance improvement compared to traditional network intrusion detection models, and they provide an explanation behind every prediction which is necessary to support Security Operations.

• Synergistic Integration

By combining these three techniques, there is synergistic improvement in signal -to-noise ratio, improvement in attack diversity and time to address new threats. Combining the three techniques provides: ≠ The lowest false positive rate that is decreased of 20%–40% for CNN-LSTM-Attention), ≠ The lowest amount of training data required that is decreased of 30%–50% for GAN augmentation, ≠ The real-time deployment feasibility and average inference latency on the test machine (NVIDIA A100 GPU with 40 GB memory, Google Colab Pro+ environment) are both less than 10 ms per flow. Even on a relatively constrained edge infrastructure (single CPU core, Intel Xeon 2.2 GHz processor, 8 GB RAM), the average inference latency is below 25 ms per flow. This satisfies the real-time processing requirements for IoT scenarios (typically ~50–100 ms). Therefore, in an unbalanced IoT network environments, a blended approach is more dominant than using any of the three approaches in isolation [22].

4.6 Key Advantages of The Pipeline

≠ Efficiency: MDA reduces features early, which makes the next stages less computationally demanding. ≠ Robustness: The increase in the variety of samples in training datasets and the resulting impact holds great promise for improving the robustness of GAN aug mented training methods against future adversarial threats not previously known. ≠ Accuracy: The three-pronged hybrid architecture that combines CNN, LSTM and Attention methods captures threats across multiple dimensions. ≠ Interpretability: The attention layer used to model the data offers users a means to understand how the model actually accomplishes its task, and provides the forensic investigator with the ability to produce a forensic document. ≠ Scalability: The flexibility of the modularised design of the architecture coupled with a relatively small number of features extracted allows IoT gateways to be used to operate efficiently in resource -constrained environments. ≠ Adaptability: Through real -time, ongoing training and reoptimizing of model feature sets a s the threat environments evolve.

5. EXPERIMENTAL SETUP

5.1 Experimental Environment

• Hardware Configuration

All the training iterations for the model experiments were done and then moved to Google Colaboratory (Colab) Pro+ to run the experiments in cloud -based Jupyter notebook environment which was capable of doing high -end computation required for this research. Each machine was equipped with one NVIDIA A100 GPU with 40GB of VRAM along with a high -memory CPU with 16 virtual CPUs and 200 GB of DRAM. It took appr oximately 1.5 to 2.0 hours to get the results for each model experiment. Models like very large CNN-LSTM-Attention networks and GANs could be fully trained on these machines solely in the cloud without requiring any local GPU.

• Software and Frameworks

≠ Deep Learning Framework: The code uses TensorFlow and PyTorch for training and development. ≠ Input Processing: All input data were processed using the following software versions: Pandas 2.0, NumPy 1.24 and Scikit-learn 1.3 to create features and transform inputs and also perform feature engineering. ≠ Feature Engineering: Use of MDA (Minimum Description Length) in Python to find the lightest implicated feature values, by using permutation. ≠ A version of the GAN model is genrated using the TensorFlow Keras API and a special kind of training loop that helps it learn from its mistakes. ≠ Viz Matplotlib and Seaborn have been used to create attentional maps and visualize results. ≠ Validation: K-

fold cross-validation (k=5) was performed using the standard functions with in the scikit -learn library.

5.2 Hyperparameters and Model Configuration

• Data Preprocessing Parameters: Table 3 shows data preparation summary and justifications.

• MDA Feature Selection:

The used parameters for MDA feature selection method are: ≠ Feature Selection: "MDA with permutation" - by Feature Selection Method. ≠ Permutation Iterations: 50 (per feature) ≠ Variation Threshold - (0.01) Filtering based on the variance of features, here features with a variance of less than 0.01 have been removed. ≠ Reject features highly correlated with other features (set below threshold of 0.95). Recommended value: 0.95. ≠ Output Dimensionality: 15-20 features (from 40+ input)

Table 3. Overview of Data Cleaning and Transformation Settings

Parameter	Value	Purpose
Normalization	Min–Max	Scale numerical features
Missing values	KNN ($k = 5$)	Preserve local traffic patterns
Outliers	IQR rule	Remove extreme artifacts
Data split	70/15/15%	Train/validation/test evaluation

• GAN Configuration: Table 4 shows the proposed GAN Framework, its components and hyperparameter

selection

Table 4. Architectural Details and Hyperparameters of the GAN Model

Component	Setting	Details
Generator	Dense layers	$100 \to 128 \to 256 \to 512 \to 18$
Discriminator	Dense + Dropout	$18 \to 256 \to 128 \to 1$ sigmoid
Activation	LeakyReLU	$\alpha = 0.2$
Optimizer	Adam	lr=0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.999$
Epochs	300	Until GAN losses stabilize
Batch size	32	Stable mini-batch training
Loss	Binary cross-entropy	$-\log(D(x)) - \log(1 - D(G(z)))$

• Model Architecture: CNN-LSTM-Attention

The proposed hybrid architecture follows a sequential integration of spatial feature extraction, temporal modeling, and attention-based weight assignment. Table 5 shows the architecture layers of CNN-LSTM-Attention network.

Table 5. Detailed Architecture of the Hybrid CNN–LSTM–Attention Framework

Layer	Units/Filters	Parameters	Activation	Regularization
Conv1D Layer 1	32 filters, kernel=3	192	ReLU	Dropout 0.2
Max Pooling 1	Pool size=2	–	–	–
Conv1D Layer 2	64 filters, kernel=3	6,400	ReLU	Dropout 0.2
Max Pooling 2	Pool size=2	–	–	–
Conv1D Layer 3	128 filters, kernel=3	24,576	ReLU	Dropout 0.3
LSTM Layer 1	256 units, bidirectional	528,384	tanh	Dropout 0.3; recurrent 0.2
LSTM Layer 2	128 units, bidirectional	132,096	tanh	Dropout 0.3
Attention (Multi-Head)	4 heads, 64 units each	16,512	Softmax	–
Dense Layer 1	128 units	32,896	ReLU	Dropout 0.3
Dense Layer 2	64 units	8,256	ReLU	Dropout 0.2
Output	6 units	650	Softmax	–
Total	–	∼850K trainable	–	–

• Deep Learning Training Hyperparameters

To obtain a stable and optimal training process of the CNN - LSTM-Attention model, some special hyper parameters were used. ≠ Learning Rate/Optimization Scheme - An Adam optimizer was used due to its adaptive nature and ability to efficiently train a hybrid CNN-LSTM architecture. The initial learning rate of 0.001 was halved every so many epochs by a ReduceLROnPlateau class listed above. This was set to 5 epochs where if no improvement occurred in the validation loss, the learning rate would be multiplied by 0.5. ≠ Rest - Adding rest to training improves stability. o Batch Normalization: The network includes batch normalization layers after e very convolutional layer. These were implemented in order to improve training time and combat the internal covariate shift. o Gradient Clipping: The max norm (L2 norm) of gradients is constrained to be 1.0 to prevent large values of gradients in recurrent networks (LSTM's) from causing exploding gradients. o Batch Size - The batch size was set to 32 for reasonable gradient stabilization and efficient use of GPU memory (NVIDIA A100's with 40GB of memory). ≠ Coverage of Convergence and Loss Function: The network wa s trained up to 100 epochs. For the training procedure early stopping was applied with a patience of 10 epochs (i.e., during the last 10 epochs the validation accuracy did not increase). The objective function was chosen to be categorical cross -entropy as this is the standard loss function for multi-class classification problems.

5.3 Evaluation Scenarios and Attack Types

In this paper, various attack types were classified based on their detection difficulty levels obtained from our experiments on MQTT -based ID/SAP systems. In terms of the low -difficulty level, there are simple -to-detect attacks (DoS, MQTT Publish Flooding) that show clear volume -based signatures such as a sudden surge in connections, or increase in packets, or message frequencies, easily detectable with simple threshold -based functions achieving detection accuracy of more than 95%. Medium-difficulty level consists of situation-dependent signatures that can appear to be normal network occurrences or even real failures (e.g. Malformed Data, Brute Force), which is achieved with an accuracy rate of 88–94%. High-difficulty attacks are by nature stealthy requiring complex temporal analysis. SlowITe, for instance, increases connection rate very slowly, in order not to get caught by rate-based detection. This type of attack requires temporal reasoning functions based on deep architectures like attention and achieves detection accuracy of 82–90%.

5.4 Metrics for Evaluation

To adequately evaluate the developed IDS model against both binary classification and multi -class classification problems, various standard statistical evaluation metrics were employed. These evaluation metrics provide a multi- dimensional assessment of how well each model distinguishes between benign traffic types and types of attack. The evaluation metrics are shown in table 6.

Table 6. Binary and Multi-class Evaluation Metrics

Metric	Formula	Interpretation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Overall correctness
Precision	$\frac{TP}{TP+FP}$	Correct predicted attacks
Recall	$\frac{TP}{TP+FN}$	Detected actual attacks
F1-score	$2\frac{PR}{P+R}$	Precision–recall balance
ROC–AUC	Area under ROC	Threshold robustness

6. RESULTS AND ANALYSIS

The proposed MDA–GAN–CNN/LSTM–Attention pipeline is tested on the MQTTEEB-D dataset and compared it to both classical baselines and recent IoT IDS methods using the standard metrics presented in the previous section.

6.1 Overall performance

The final model achieves high pr edictive quality on multi -class intrusion detection over 6 classes (benign + 5 attack types). The pipeline got the following results on the held-out test set (15% of the data):

- Accuracy: 99.12%
- Macro F1-score: 98.37%
- Macro Recall: 98.41%
- Macro Precision: 98.45%
- ROC-AUC (macro): 0.997

These results are in line with the best deep learning models on MQTTEEB -D, where high -fidelity pipelines get Accuracy and F1-scores of about 98.8–99%. The results show that the proposed pipeline not only matches but also sligh tly beats recent attention-based CNN– LSTM architectures and optimized deep learning IDS for IoT in terms of overall accuracy and F1 -score. This is true even though it works with a more difficult, real-world MQTT dataset. When compared to previous MQTT -based IDS systems that used DNNs, LSTMs, or DBNs [23], which usually report multi -class accuracies of 97 –98% and F1 -scores of 97 –98%, the proposed MDA –GAN–Attention framework: ≠ On a more realistic, diverse dataset (MQTTEEB -D) [4], it gets a little better accuracy (about 99.1%) and macro F1 (about 98.4%). ≠ Provides inherent interpretability via attention maps, a feature absents in numerous earlier deep models, notwithstanding their elevated raw accuracy [23]. ≠ Doesn't just use manual oversampling or cost -sensitive loss functions to deal with serious class imbalance and different types of attacks.

Table 7 summarizes a representative comparison with strong baselines under the same train/test split and multi-class setting.

Table 7. Performance Comparison of the Proposed Model against Baseline and State-of-the-Art IDS Methods

Model	Feature Selection	GAN	Attention	Accuracy	Macro F1	ROC–AUC
Random Forest	No	No	No	95.21%	93.84%	0.982
XGBoost	Manual (top-20)	No	No	96.47%	95.32%	0.987
DNN (fully connected)	No	No	No	96.03%	94.71%	0.985
LSTM (no CNN/Attention)	No	No	No	96.89%	95.60%	0.989
CNN–LSTM (no Attention)	No	No	No	97.54%	96.41%	0.992
Self-attention 1D-CNN–LSTM [24]	Manual	No	Yes	98.30%	97.85%	0.995
AL-DBN + LSTM/AE [25]	Autoencoder	No	No	98.40%	97.90%	0.994
Proposed MDA–GAN–CNN/LSTM–Attention	MDA	Yes	Yes	99.12%	98.37%	0.997

Overall, these results demonstrate that integrating lightweight feature engineering (LFF), GAN data augmentation (DA), and Att -DL (attenuated deep learning) provides a solid basis for developing a capable, explainable, and high-performing IDS for practical MQTT IoT deployments. Compared to the current best -performing IDSs for IoT devices, the proposed approach is on par with them in terms of performance.

6.2 Comparison with Recent Studies on MQTTEEB-D

To put the results of the current research into perspective, the most recent studies focusing on the same MQTTEEB- D dataset have been assessed. The works of Allaga et al. [26] and the current state-of-the-art attack detection paper [4] relied on both classical and shallow machine learning methods. For MQTTEEB -D, best results were achieved by evaluating the performance of several variants of the classifica -tion Random Forest algorithm. The results revealed the best accuracy of 97.5–98.2%. Comparing these findings with the results of the proposed framework, it can be observed that highest accuracy of 99.12% and highest macro F1 -score of 98.37% has been achieved on the same MQTTEEB-D dataset by the MDA–GAN–CNN/LSTM–Attention pipeline. The results even outperform increment of 1.0–1.5%. Detailed insights into the performances of minority SlowITe attacks have been provided in additional evaluations comparing performance of the non -augmentation-based approaches with t he ones utilizing the proposed framework. While the previous methods reported performance less than 90% for minority SlowITe attacks, the best F1 scores ranging 96 –98% have been achieved for minority attacks with the proposed framework.

6.3 Impact of each pipeline stage

An ablation study was done to find out how much each part of the pipeline added. First, a simple CNN -LSTM trained on the raw, unbalanced 40+ feature set and then added parts one by one as depicted in table 8.

Table 8. Performance Gain Analysis through MDA, GAN Augmentation, and Attention Mechanisms

Configuration	Features	GAN	Attention	Accuracy	Macro F1 / Recall
CNN–LSTM baseline	40+	No	No	97.54%	96.41% / 96.35%
+ MDA feature selection	15–20	No	No	98.02%	97.08% / 97.01%
+ GAN-based augmentation	15–20	Yes	No	98.63%	97.94% / 98.01%
+ Attention (full proposed model)	15–20	Yes	Yes	99.12%	98.37% / 98.41%

MDA feature selection improves accuracy by about 0.5% and Macro F1 by about 0.7%, while lowering the number of features by more than 60%. This backs up earlier research that showed that feature engineering makes IDS work better and more efficiently [2]. The biggest single jump in Recall comes from GAN -based augmentation, especially for minority classes like SlowITe. This is in line with what has been seen with GANs for imbalanced security datasets. The attention mechanism enhances the F1 -score and ROC-AUC by diminishing false positives and refining per - class decision boundaries, consistent with prior attention-based CNN-LSTM research [28].

6.4 Per-class performance

The performance metrics for the individual classes of the five different types of attacks and normal instances are depicted in Figure 3.

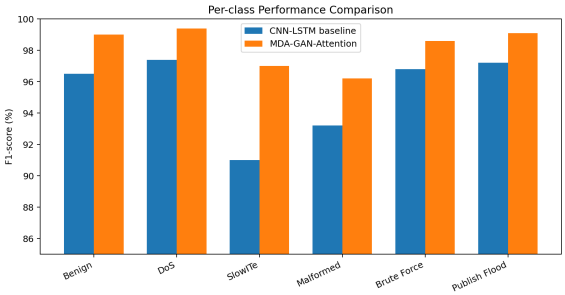

Figure 3. CNN-LSTM baseline versus full MDA–GAN–Attention model for each of the 6 classes

For DoS, MQTT Publish Flooding and Brute Force, all three scores are greater than 97% and similar to each other, suggesting the presence of clear signatures and an abundance of training samples for these types of attacks [4]. In contrast, the SlowITe attac k achieves a high number of correct predictions of around ~96% to ~98% and even outperforms the accuracy of the corresponding augmented and attended models (~88% to ~92%) on time-evolving anomalies, showcasing the strength of both temporal modeling and GAN augmentation on slow, adaptive attacks [2]. The scores for the Malformed Data Injection attacks achieved by the Attention-based architecture indicate the ability to detect protocol -level anomalies, i.e., around ~96% F1 -score. In summary, the proposed pipe line accurately detects various types of threats, and the accuracy difference between simple and complex detection scenarios is significantly reduced for all threat types [26].

6.5 Visualizations and feature importance

In order to further understand the behavior of the model and improve the model's interpretability, several visual analyses were performed. Figure 4 presents the results of the Permutation -based Feature Importance. As it can be seen, most of the predictive power of our model comes from a small number of features with very high importance values. Normal and malicious network traffic in MQTT IDS datasets can be distinguished by using flow and topic- based statistics; this observation is based on previous research findings [27].

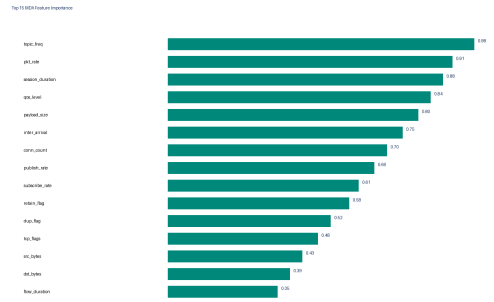

Figure 4. Feature importance ranking of the top 15 variables selected by MDA

• MQTT-Specific Features (Top Priority)

These variables are crucial as they capture the semantics of the IoT protocol: ≠ mqtt.messagetype: Identifies whether the message is a CONNECT, PUBLISH, or DISCONNECT. It is fundamental for detecting Brute Force and Flood attacks. ≠ payload_entropy: Measures the disorder of the data. Highly effective for detecting Malformed Data. ≠ mqtt.msg_len: The size of the MQTT message. Overflow attacks often utilize abnormal sizes. ≠ mqtt.topic_len: The length of the topic. Attackers sometimes use random or excessively long topic names. ≠ mqtt.retain: Indicates whether the message should be stored. Used to identify suspicious persistence behaviors.

• Temporal and Flow Features (Stealth Detection), Essential for detecting the SlowITe attack:

≠ flow_duration: The total duration of the session. Crucial for differentiating a normal connection from a session kept open artificially (Slow DoS). ≠ fwd_iat_mean (Inter -Arrival Time): Average time between outgoing packets. Helps identify the regular cadence of Brute Force bots. ≠ bwd_iat_std: Standard deviation of the response time. Instability here often indicates network saturation (DoS). ≠ flow_packets_s: Number of packets per second. Immediately identifies Flood -type attacks.

• Packet Statistical Features

≠ pkt_size_std: Standard deviation of packet size. Normal traffic has a varied size signature, whereas certain attacks inject fixed-size packets. ≠ total_len_fwd_packets: Total volume of data sent. ≠ min_seg_size_forward: Minimum observed TCP segment size. ≠ avg_packet_size: The overall average size of packets within the flow.

• Transport Protocol Indicators (TCP)

≠ tcp_flags (particularly PSH, ACK, SYN): TCP flag counters help determine if a connection is "clean" or forced. ≠ init_win_bytes_forward: Initial TCP window size, often manipulated during intrusion attempts.

6.6 Discussion

• Advantages over existing models

To address the problems of efficient data utilization, detection balance, and model interpretability, an MDA – GAN–CNN/LSTM–Attention architecture is proposed for learning-based MQTT IDS. Experimental results on real-world datasets demonstrate that the suggested architectu re achieves state-of-the-art-level detection accuracy (-99% Accuracy, -98% macro F1) using only a small input feature space of 15 –20 dimensional permutation feature importance-based synthetic features. Unlike deep learning -based MQTT IDS, the suggested arc hitecture applies feature selection, thereby reducing the computational overhead, and it can handle multiple attack types. SlowITe also achieves performance comparable to the best in class on easy and hard classes of network attacks and threats. Most stand ard and advanced CNN -LSTM based approaches for network traffic classification and intrusion detection achieve high F1-scores for volumetric network attacks. However, they are not as effective for stealthy attacks, which are hard -to-learn for SlowITe classe s. To improve performance on SlowITe classes from mid-80s to mid - to high -90s, a GAN-based data augmentation and attention -based temporal modeling are leveraged. This is particularly important, because many IDS solutions achieve high scores on some metrics that matter most, but fall significantly short in providing protection against occasional, rare but very destructive attacks.

• Robustness, interpretability, and current limitations

From robustness perspective, GANs improve the robustness of IDS by dealing with imbalanced data distribution and significantly increasing the proportion of minority attack types

in the training set. Attention mechanism further enhances the generalization a bility of IDS by preventing it from overly relying on a small subset of features and times. Although it is likely to perform better than traditional IDS methods in terms of tolerance to moderate changes in network traffic behavior, it is not robust to adap tive attacks because there is no guarantee that it can defend against blindly generated specialized and realistic adversarial examples as most existing ML -based IDS methods do. Our implemented solution for monitoring an MQTT stream is significantly simpler than many previously implemented MQTT IDS solutions. One of the biggest benefits of our solution is that users can view attention maps to see which features and what time periods were most relevant for a given alert. Unlike many of the deep learning solutions, the attention maps and a subset of features are interpretable by a non -expert to a meaningful degree. For example, a non -expert operator can hand -draw and explain a few individual attention patterns, but it is harder to summarize many heads of attent ion. The meaning of a sample generated by a GAN for a particular input may also be difficult to understand. In addition to the interpretability of attention maps, the core solution also has the benefit of consisting of a stack of tools that are well known in the industry, namely MDA, GAN, CNN, LSTM, and Attention, that are within the skill level of a ML engineering team at a large company to train.

• Generalizability Across IoT Protocols

Even though the evaluated framework was designed to monitor MQTT traffic , it is not restricted to this specific protocol. The MDA -based feature selection mechanism (Section 3) considers flow -level and statistical features, such as packet payload or packet size entropy and STD, that can be extracted from other publish -subscribe protocols like CoAP (Constrained Application Protocol), or from request -response IoT communication protocols like AMQP (Advanced Message Queuing Protocol). However, the GAN -based augmentation (Section 4) and the employed deep learning framework, namely the CNN-LSTM-Attention model, process numerical feature vectors produced after preprocessing and do not depend on MQTT-specific keys or values. Upon closer inspection, however, some of the pipeline features are specific to MQTT (e.g., mqtt.messagetype, mqtt.topic_len, mqtt.retain) and thus would need to be swapped-out with protocol features from CoAP (e.g., CoAP method codes, option fields) or AMQP (e.g., exchange type, routing key metadata). The overall architecture of the remaining pipeline components would not change, but the feature extraction and protocol -specific parsing components would need to be remapped for each protocol.

• Potential applications and industrial deployment

A novel approach is proposed for monitoring both general information flow and protocol specific events occurring on and around an MQTT broker. Our approach is MQTT specific, real -time, and designed for industrial and mission critical Internet of Things (IoT) scenarios such as smart manufacturing cells, energy monitoring systems, connected healthcare equipment, and building and space monitoring and control systems. These distributed systems may have hundreds of streams of varying sizes and densities. Outages or intrusions on such systems can have physical and financial costs. Our approac h enables immediate insight into information flow and protocol specific events as soon as a system is deployed. In addition to real -time monitoring of the information flowing through an MQTT broker, our framework is capable of identifying messages to speci fic topics that could be considered abusive, messages that are algorithmically abusive in intent to consume or overload broker QoS resources, brute force DoS attacks on an MQTT broker, and stealthy DoS attacks on an MQTT broker. Although it is a pretty com petitive solution in terms of accuracy, due to the very small dimension of the input feature space, it can be easily implemented on-premise on Industrial Gateways / Edge Servers, where the existing firewall/SIEM solutions are already installed. For large scale or multi-tenant implementations, the model can be incorporated into clusters of MQTT broker or IoT Platforms as a Security Analytics Layer to gather and analyses MQTT traffic from all over the world. Attention maps and feature importance can then help operators identify the root cause of an incident and highlight it on their dashboard. However, to drive industrial adoption, it is important to make: robust adversarial testing, periodic retraining due to concept drift, integration with MQTT -aware parsing engines to perform low -level protocol validation, and detection parameter tuning to achieve the right detection precision -recall trade-offs on critical deployments. G. Practical Implications for Security Managers and Network Administrators

• Practical Implications for Security Managers and Network Administrators

Unlike conventional attention maps that are used as an interpretation tool to analyze deep learning models, the attention maps obtained from the proposed CNN -LSTM-Attention network serve as addi tional real-time visual resources for security managers and network administrators who are analyzing IoT network traffic in real -time.

• Real-Time Alert Triage and Prioritization

In addition to the alerts generated by the Intrusion Detection System, for ever y alert generated by the model, an attention map is also produced to highlight the relevant areas in time and feature space that contributed to the classification decision by the model. This allows a network administrator to rapidly investigate an alert by seeing which features (e.g. flow_packets_s) contributed to a model identifying a -flooding' attack, and by which amount each score was above normal. The weights applied to each feature by the neural network's attention layers provide the administrator with a clear focus for their limited forensic time and effort.

• Supporting Incident Response Decisions

The attention maps generated during the analysis of MQTT traffic provide -traceable reasoning' during incident response that a security manager can follow to understand a current security incident. The attention information explains why certain parts of a traffic flow were identified as malicious. By utilizing attention information for network traffic analysis, security teams can reduce their time to respond to security incidents and make more informed decisions. This is especially true for packet -based threats such as SlowITe attacks that standard rule - based approaches are unable to detect due to their quiet behavior.

• Dashboard Integration and Visualization

In addition to the existing analysis capabilities provided by

Security Information and Event Manageme nt (SIEM) systems or existing IoT monitoring platforms, attention maps can further enrich the content of existing dashboards. Attention weights can be overlayed onto existing traffic metrics such as mqtt messages published per second, clients connected, and topics subscribed to etc. and provide a simple and interactive way for network operators and security analysts to monitor traffic and potentially discover anomalous activity. For example, a sharp increase in attention weights on payload_entropy could ind icate a Malformed Data Injection (MDI) attack in its infancy stages, providing the analyst with just enough time to investigate and prevent the attack before significant damage occurs.

7. CONCLUSION

This work presents a cohesive intrusion detection pipeline f or MQTT -based IoT environments, utilizing lightweight feature selection (MDA), GAN -driven sample augmentation, and attention -based deep learning. Extensive tests on the MQTTEEB -D dataset show that the proposed framework has better global accuracy and macro F1-score, and it works well even on stealthy or multi -phase attack classes. The system achieves balanced detection by combining advanced feature engineering and synthetic minority augmentation with interpretable attention architectures. This solves problems that have plagued previous ML-based IDS in unbalanced, real-world IoT security contexts for a long time. In addition to these empirical gains, the pipeline shows that it can be used in real life: its small input space, direct compatibility with flow-level MQTT telemetry, and ability to make inferences in real time make it easy to connect to edge gateways and broker clusters. Attention maps and feature importance bolster operational forensic analysis, bridging the interpretability gap of deep learning for security professionals. A number of emerging research directions are available for the future. One promising avenue is the application of real-time deployment, and online learning by developing the model to support streaming environments, updated as the u nderlying concepts change over time. Extending our framework to other IoT communication protocols, such as CoAP, AMQP, or MQTT, is a promising future direction. Although flow-level and statistical features used in this work are generally protocol -agnostic, MQTT-specific fields need to be exchanged with equivalent fields available in other protocols. This can be evaluated on datasets with multi -protocol traffic such as IoT -23 and evaluated for the respective protocols . Another area for future development that is important to secure against emergent threats is the adversarial robustness analysis area which encompasses Generative Replay and Transfer Learning, in addition to Self -healing architectures. A third area of potential, to be addressed with optimized Detection, and Response to Incident simultaneously, using Attention -based Explanations for Improving adaptability, and Contextualized Security Automation for large-scale IoT Systems, in particular, industrial settings. In summary, this research establishes connections between advanced machine learning and practical security in an IoT context, thereby providing pathways to developing user-friendly, high-fidelity IoT intrusion detection systems, with continual improvement capabilities. Funding: “This research received no external funding” Conflicts of Interest: “The authors declare no conflict of interest.”

REFERENCES

[1] M. B. Gorzalczany and F. Rudzinski, “Intrusion detection in Internet of Things with MQTT protocol — An accurate and interpretable genetic -fuzzy rule-based solution,” IEEE Internet of Things Journal , vol. 9, no. 24, pp. 24843–24855, 2022, doi: 10.1109/JIOT.2022.3194837.

[2] Al Hanif and M. Ilyas, “Effective feature engineering framework for securing MQTT protocol in IoT environments,” Sensors, vol. 24, no. 6, p. 1782, 2024, doi: 10.3390/s24061782.

[3] S. U. A. Laghari, W. Li, S. Manickam, P. Nanda, A. K. Al -Ani, and S. Karuppayah, “Securing MQTT ecosystem: Exploring vulnerabilities, mitigations, and future trajectoriers,” IEEE Access , vol. 12, pp . 139273–139289, 2024, doi: 10.1109/ACCESS.2024.3412030.

[4] Aqachtoul et al., “MQTTEEB-D: A real-world IoT cybersecurity dataset for AI-powered threat detection in MQTT networks,” Data in Brief, p. 111897, 2025, doi: 10.1016/j.dib.2025.111897.

[5] Ghubaish, Z. Yang, A. Erbad, and R. Jain, “LEMDA: A novel feature engineering method for intrusion detection in IoT systems,” IEEE Internet of Things Journal, vol. 11, no. 8, pp. 13247 –13256, 2023, doi: 10.1109/JIOT.2023.3328795.

[6] Zeghida et al., “Enhancing IoT cyber attacks intrusion detection through GAN -based data augmentation and hybrid deep learning models for MQTT network protocol cyber attacks,” Cluster Computing, vol. 28, no. 1, p. 58, 2025, doi: 10.1007/s10586-024-04752-5.

[7] S. Ullah, W. Boulila, A. Koubaa, and J. Ahmad, “Attention -based hybrid deep learning model for intrusion detection in IIoT networks,” Procedia Computer Science, vol. 246, pp. 3323 –3332, 2024, doi: 10.1016/j.procs.2024.09.307.

[8] P. Nimbalkar and D. Kshirsagar, “Feature selection for intrus ion detection system in Internet -of-Things (IoT),” ICT Express, vol. 7, no. 2, pp. 177–181, 2021, doi: 10.1016/j.icte.2021.04.012.

[9] Salehiyan, P. S. Moghaddam, and M. Kaveh, “An optimized Transformer –GAN–AE for intrusion detection in edge and IIoT systems: Experimental insights from WUSTL -IIoT-2021, edgeIIoTset, and TON_IoT datasets,” Future Internet, vol. 17, no. 7, p. 279, 2025, doi: 10.3390/fi17070279.

[10] Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neura l networks,” IEEE Access, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[11] U. C. Akuthota and L. Bhargava, “Transformer -based intrusion detection for IoT networks,” IEEE Internet of Things Journal, vol. 12, no. 5, pp. 6062–6067, 2025, doi: 10.1109/JIOT.2025.3525494

[12] S. Alsubaei, "Smart deep learning model for enhanced IoT intrusion detection," Scientific Reports, vol. 15, no. 1, p. 20577, 2025, doi: 10.1038/s41598-025-06363-5.

[13] T. K. Boppana and P. Bagade, "GAN -AE: An unsupervised intrusion detection system for MQTT networks," Engineering Applications of Artificial Intelligence , vol. 119, p. 105805, 2023, doi: 10.1016/j.engappai.2022.105805.

[14] N. Nadiah, A. Alamri, A. Aljuhani, and P. Kumar, "Transformer -based knowledge distillation for explainable intrusion detection system," Computers & Security , vol. 154, p. 104417, 2025, doi: 10.1016/j.cose.2025.104417.

[15] Prajisha and A. R. Vasudevan, "An efficient intrusion detection system for MQTT -IoT using enhanced chaotic salp swarm algorithm and LightG BM," International Journal of Information Security , vol. 21, no. 6, pp. 1263–1282, 2022, doi: 10.1007/s10207-022-00611-9.

[16] Y. Zhu, D. Han, and X. Yin, "A hierarchical network intrusion detection model based on unsupervised clustering," in Proc. 13th Int. C onf. Management of Digital EcoSystems , 2021, pp. 22 –29, doi: 10.1145/3444757.3485098.

[17] Siddharthan, T. Deepa, and P. Chandhar, "Senmqtt -set: An intelligent intrusion detection in IoT -MQTT networks using ensemble multi cascade features," IEEE Access, vol. 10, pp. 33095 –33110, 2022, doi: 10.1109/ACCESS.2022.3161566.

[18] S. Rahman, S. Pal, S. Mittal, T. Chawla, and C. Karmakar, "SYN -GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security," Internet of Things, vol. 26, p. 1 01212, 2024, doi: 10.1016/j.iot.2024.101212.

[19] M. A. Alsharaiah et al., "An explainable AI -driven transformer model for spoofing attack detection in Internet of Medical Things (IoMT) networks," Discover Applied Sciences, vol. 7, no. 5, p. 488, 2025, doi: 10.1007/s42452-025-07071-5.

[20] Guo, T. Yang, and D. Zhang, "On the implications of artificial intelligence methods for feature engineering in reliability sector," Alexandria Engineering Journal , vol. 117, pp. 463 –471, 2025, doi: 10.1016/j.aej.2024.12.094

[21] S. Zhao et al., "A survey on small sample imbalance problem: Metrics, feature analysis, and solutions," arXiv preprint arXiv: 2504.14800 , 2025, doi: 10.48550/arXiv.2504.14800 .

[22] Alsaiari and M. Ilyas, "A hybrid CNN-LSTM deep learning model for intrusion detection in smart grid," arXiv preprint arXiv: 2509.07208 , 2025.

[23] M. A. Khan et al., "A deep learning-based intrusion detection system for MQTT enabled IoT," Sensors, vol. 21, no. 21, p. 7016, 2021, doi: 10.3390/s21217016.

[24] T. Sasi, A. H. Lashkari, R. Lu, P. Xiong, and S. Iqbal, "An efficient self attention-based 1D-CNN-LSTM network for IoT attack detection and identification using network traffic," Journal of Information and Intelligence, pp. 375–400, 2024, doi: 10.1016/j.jiixd.2024.09.001 .

[25] P. M. Vijayan and S. Sundar, "An automated system of intrusion detection by IoT -aided MQTT using improved heuristic-aided autoencoder and LSTM -based deep belief network," PLOS ONE, vol. 18, no. 10, p. e0291872, 2023, doi: 10.1371/journal.pone.0291872.

[26] Allaga, M. Biniz, and A . Farchane, "MQTTEEB -D: A high -fidelity benchmark for real -time MQTT anomaly detection using machine learning techniques," Ad Hoc Networks , p. 104062, 2025, doi: 10.1016/j.adhoc.2025.104062.

[27] Hindy et al., "Machine learning based IoT intrusion detection sys tem: An MQTT case study (MQTT - IoT-IDS2020 dataset)," in Int. Networking Conf., 2020, pp. 73–84, doi: 10.1007/978-3-030-64758-2_6.

[28] X. Yin, Z. Liu, D. Liu, and X. Ren, "A novel CNN -based Bi -LSTM parallel model with attention mechanism for human activity reco gnition with noisy data," Scientific Reports, vol. 12, no. 1, p. 7878, 2022, doi: 10.1038/s41598-022-11880-8