AI-Blockchain Approach for MQTT Security: A Supply Chain Case Study

Raouya AKNIN, Hind El Makhtoum, Youssef Bentaleb
Ibn Tofail University, Engineering Sciences Laboratory, Kenitra, Morocco
0000-0002-4644-055X

Abstract—**The use of the MQTT protocol in critical sectors such as healthcare and industry has prompted research to propose solutions for strengthening its security and preventing it from attacks that are growing exponentially and becoming increasingly sophisticated and difficult to detect. This paper aims to improve the security of the MQTT architecture, ensuring it is resilient to current attacks and adaptable to potential future attacks while considering the constraints of the IoT environment. To achieve this, the proposed architecture is based on the interaction between the AI model, which continuously analyzes device behavior, and smart contracts, which automatically apply appropriate security measures once fraud is detected. A device reputation mechanism is used to prevent malicious devices from rejoining the network. The AI model proposed in this article was initially trained on a set of malicious behaviors using supervised learning. The results show that the detection accuracy achieved 95.97%. This accuracy is expected to improve over time through the integration of unsupervised learning into the architecture, enabling the discovery of new attack patterns and additional parameters for malicious behavior identification. For simulation testing, the architecture was applied to supply chain management as a case study of critical applications, and smart contracts were deployed in the Remix environment. The architecture demonstrated resilience and robustness across various attack scenarios.**

Keywords—IoT; MQTT; blockchain; smart contracts; AI hybrid model; device reputation

I. Introduction

The advent of the Internet of Things (IoT) has revolutionized the interaction between real objects and the digital world by breaking down the boundaries between these two worlds. Indeed, The emergence of cutting-edge technologies such as wireless sensor networks (WSN), Radio Frequency Identification (RFID), cloud computing, and others that facilitate real-time data collection, sharing, and analysis has enabled better real-time decision-making, remote supervision and control of environments, and the automation of tasks and processes. This has opened up a wide range of IoT advanced services in several domains ranging from simple applications such as smart household appliances to critical applications like supply chain management, smart grids, and healthcare applications. According to [1], by the end of 2025, the number of IoT devices worldwide is expected to exceed 75 billion devices and the compound annual growth (CAGR) in the IoT market is forecast to average 10.49% between 2024 and 2029, bringing the total market value to $1,560 billion by 2029 [2].

Nevertheless, the expansion of connected objects into more sensitive and critical areas has raised concerns about the security of the protocols frequently employed in this environment. The native security measures of these protocols no longer meet the requirements of critical applications, and attacks targeting the IoT environment have become more numerous and complex. For instance, the Message Queuing Telemetry Transport (MQTT), the most widely used protocol in this environment, is vulnerable to several types of attacks, including man-in-the-middle (MITM) attacks due to a lack of encryption, denial-of-service (DoS) attacks because of the centralized architecture and the broker's single point of failure, and unauthorized access resulting from weaknesses in the authentication mechanism.

To overcome these challenges, many articles have proposed an enhanced security architecture using blockchain as the one of the promising solutions [3]–[8]. Indeed, the decentralized nature of blockchain minimizes denial-of-service attacks and avoids the single points of failure, which are considered nightmares in the field of cybersecurity. Moreover, it ensures data integrity and enables the traceability of actions carried out on the network. The use of smart contracts to automate actions on the network is another advantage of using blockchain technology. Artificial intelligence (AI) is also another a cutting-edge technology that is used to enhance the MQTT architecture by proposing solutions to detect the malicious behaviors and potential MQTT attacks [9]–[11].

The goal of this article is to propose an MQTT architecture that meets the security requirements of critical IoT applications, is resilient to current attacks, and is adaptable to potential attacks, while taking into account the IoT constraints. To this end, it aims to improve the architecture proposed in our previous research work [4] by adding an additional layer of protection based on artificial intelligence. The advantage of the proposed solution is that it combines AI technology to detect the abnormal behavior of connected devices or those attempting to connect to the network, with smart contracts to automatically apply the appropriate security measures. It also introduces the concept of device reputation to prevent malicious devices from rejoining the network.

The remainder of this paper is structured as as follows: Section II provides a summary of research works that have been proposed improved MQTT architectures using cutting-edge technologies. Section III outlines the paper's contribution by combining blockchain and AI technologies to enhance MQTT protocol security. Section IV presents our proposed solution that combines blockchain and AI for a resilient and attack-resistant MQTT architecture. Section V discusses the results of attack test scenarios on the proposed solution. Section VI summarizes the main ideas discussed in this paper and provides an overview of future research directions to further improve the security of the MQTT protocol while respecting the constraints of the IoT environment.

II. Related Works

Several research works have proposed solutions to improve the security of the MQTT protocol, which is the most widely used protocol in the IOT environment to meet the IOT critical applications requirements. Indeed, many articles have used blockhain [3]–[8] to enhance the MQTT security since it ensures data integrity, avoids a single point of failure, and enables the traceability of the actions performed in the network. Moreover, the use of smart contracts to automate network actions is another advantage of blockchain technology. On the other hand, several articles have leveraged artificial intelligence (AI) to enhance the security of the MQTT protocol. However, these studies primarily focus on attack detection. Indeed, article [9] has proposed an optimized model for intrusion detection in the MQTT-based IoT networks. For this purpose, an empirical comparison between 22 machine learning (ML) algorithms was established, concluding that the Generalized Linear Model (GLM) classifier with the random oversampling technique showed the best detection performance. Along the same lines, article [10] has proposed an AI model based on supervised learning to detect attacks threatening the MQTT protocol. To achieve this, a comparative analysis of various supervised learning algorithms was conducted. It ultimately concluded that the convolutional long short-term memory neural network (CNN-LSTM) algorithm outperforms other models in terms of accuracy and performance for intrusion detection on the MQTT protocol. Although these articles contribute to the selection of the most effective learning algorithms for MQTT attack detection, they have however, proposed AI models based on supervised learning algorithms, making them limited to the attacks specified in the training phase. To address this, article [11] has introduced an MQTT intrusion detection system based on a Generative Adversarial Network-based autoencoder (GAN-AE), an unsupervised learning algorithm that allows the detection of different types of attacks by analyzing the behavior.Although the solution showed its ability to adapt to new and evolving attack scenarios and the overall detection rate reached 99.2%, however, like the above-mentioned articles, the proposed system is limited to intrusion detection without implementing security measures to prevent these attacks. Additionally, since the proposed solution may introduce false positives, it will be difficult to implement security measures automatically. Combining AI solutions for attack detection and blockchain, more specifically, smart contracts to automatically apply appropriate security measures seems to be a promising solution. Article [1] has proposed a framework that combines AI and blockchain technologies to enhance security in the IOT environment. It has introduced a new security layer, integrating blockchain and AI into the traditional three-layer IoT architecture [12]. However, the article does not propose any real implementation and it emphasizes the need for further research to develop effective strategies for combining these technologies to create reliable and secure digital ecosystems in the IoT landscape. The research in [13] has proposed an intelligent architecture that detects smart meter authentication fraud and consequently prevents their access to the network. In addition, it introduced the notion of reputation to prevent malicious smart meters from rejoining the network. Unlike previous articles, this paper implements smart contracts to automatically apply the appropriate security measures when fraud is detected. However, the paper only proposes a theoretical solution for the IOT architecture, without specifying the protocol used or implementing the AI model. Moreover, it focuses only on authentication attacks. Table I summarizes the approaches used in related works, focusing on the contribution of the articles to enhancing the security of the MQTT protocol to meet the requirements of critical applications, the technologies used, and the limitations of the proposed solutions.

Limits and Issues: By analyzing the solutions presented in Table I, two major challenges can be identified:

- Although solutions based on blockchain and smart contracts meet the security requirements of IoT critical applications in terms of confidentiality, integrity, and availability, they cannot address all types of attacks or predict potential ones.
- Solutions designed to detect MQTT attacks are either based on supervised learning, which is limited to the attacks specified during the training phase, or on unsupervised learning, which may introduce false positives.

Combining these two promising solutions can significantly enhance MQTT security by leveraging AI techniques to detect attacks and utilizing smart contracts to enforce appropriate security measures. However, existing articles addressing this combination mainly propose theoretical models without implementing the full process. To this end, the contribution of this paper lies in implementing and testing the interaction between an AI model for attack detection and smart contracts to apply the corresponding security measures. Additionally, the solution relies on a hybrid AI model that combines a supervised learning algorithm to detect known attacks and an unsupervised learning algorithm to identify potential new or unknown attack types.

III. Paper's Contribution

The solution proposed in this article combines the use of blockchain and AI technologies to improve the security of the MQTT protocol. Indeed, the AI model allows the real-time detection of malicious behaviors and consequently the automatic application of the appropriate security measures using smart contracts, ensuring a fast and efficient response to potential attacks. Furthermore, the decentralized nature of the blockchain minimizes the risk of denial-of-service attacks and eliminates the challenges associated with the single point of failure. Blockchain also guarantees data integrity and ensures accountability for actions performed in the network, which is useful for monitoring and post-incident analysis. Moreover, the hybrid approach of the AI model enables continuous learning of sophisticated and complex attack patterns and relevant parameters for identifying malicious devices. For added security, the concept of reputation has been introduced to maintain records of device behavior and therefore prevent previously classified malicious devices from rejoining the network. Although the solution used consistent technologies such as blockchain and AI algorithms, it does not adversely affect the performance of the constrained IoT environment since resource-intensive operations are managed on the brokers network side, typically located in the cloud or data centers while the only operation executed on the device side is the OTP calculation.

TABLE I. Summarization of Approaches used in Related Works for Enhancing MQTT Security

Article	Main objective	Technology	Limitations
[4]	Proposes a decentralized MQTT architecture that meets the security requirements of critical applications without affecting the overall protocol performance or the constraints of the IoT environment.	Blockchain and smart contracts	Cannot cover all types of attacks or predict potential ones.
[5]	Improves the authentication process using a one-time password (OTP) and smart contracts to provide an independent channel for managing two-factor authentication. The solution also ensures accountability through blockchain.	Blockchain and smart contracts	• Does not address the single point of failure issue as it relies on a broker. • Focuses only on the authentication process. • Cannot cover all types of attacks or predict potential ones.
[6]	Proposes a novel approach solution that relies on blockchain sharding to achieve robust security while minimizing computational overhead.	Blockchain and smart contracts	Cannot cover all types of attacks or predict potential ones.
[7]	Proposes a decentralized solution that meets the security requirements of critical IoT applications regarding confidentiality, integrity, and control access to the topics managed by the broker.	Blockchain and smart contracts	• Does not address the single point of failure issue as it relies on a broker. • Focuses only on the authentication process. • Cannot cover all types of attacks or predict potential ones.
[8]	• Proposes a holistic, decentralized solution for securing MQTT. • Introduces a token stored on the blockchain to control topic access and avoid permanent credentials.	Blockchain and smart contracts	• Uses TLS in resource-constrained environments. • Impacts the overall protocol performance. • Cannot cover all types of attacks or predict potential ones.
[9]	Proposes an optimized model for intrusion detection in the MQTT-based IoT networks.	AI model based on supervised learning algorithm: Generalized Linear Model (GLM) classifier with the random oversampling technique.	• Provides a solution for detecting attacks without implementing security measures. • Supervised learning models are limited to attacks specified in the training phase.
[10]	Proposes an AI model based on supervised learning to detect attacks threatening the MQTT protocol.	AI model based on supervised learning algorithm: The convolutional long short-term memory neural network (CNN-LSTM).	• Provides a solution for detecting attacks without implementing security measures. • Supervised learning models are limited to attacks specified in the training phase.
[11]	• Proposes an MQTT intrusion detection system based on a Generative Adversarial Network-based auto-encoder (GAN-AE) to detect various types of attacks by analyzing behavior. • The solution demonstrated adaptability to new and evolving attack scenarios with an overall detection rate of 99.2%.	AI model based on unsupervised learning: Generative Adversarial Network based auto-encoder (GAN-AE)	• Provides a solution for detecting attacks without implementing security measures. • May introduce false positives.
[1]	Proposes a framework combining AI and blockchain technologies to enhance security in IoT by adding a new security layer to the classical three-layer architecture.	• Blockchain and smart contracts. • AI technology.	The framework is theoretical and lacks real-world implementation.
[13]	• Proposes an intelligent architecture that detects smart meter authentication fraud and consequently prevents their access to the network. • Introduces the notion of reputation to prevent malicious smart meters from rejoining the network.	• Blockchain and smart contracts. • AI technology.	• Doesn't implement the AI model. • Focuses only on authentication attacks.

IV. The Enhanced MQTT Protocol

Since attacks targeting connected objects in general, and the MQTT protocol in particular, are constantly evolving and it is impossible to anticipate and cover all potential threats, this article aims to enhance the architecture proposed in our previous work [4] by introducing a new layer of protection based on artificial intelligence. As depicted in Fig. 1, the basic architecture is based on a consortium blockchain and smart contracts to automate the MQTT authentication, publication, and subscription processes. It comprises a client, which can be a publisher or a subscriber, and a brokers network that executes the smart contracts. Communication between the components is divided into three phases: the registration phase, the connection phase, and the publication phase. The AI solution proposed in this paper aims to analyze the behavior of devices connected to the network, as well as those attempting to authenticate, in order to effectively interrupt or prevent malicious connections accordingly. The behavior analysis is based on a set of relevant parameters that enable the identification of the device's behavior to determine whether it is malicious or legitimate. Once a device is detected as malicious, its reputation is automatically changed to false to prevent its reintegration into the network. It is important to note that the device's reputation can be changed either when it matches a predefined rule in the smart contracts (wrong One-Time-Password (OTP), unregistered device, unauthorized publication, etc.) or when the AI model detects a malicious behavior. Despite the use of technologies such as blockchain and intelligent algorithms, which require the use of resources, the solution aims to account for IoT environment constraints by offloading resource-intensive operations to the brokers' network, typically located in the cloud or data centers, while leaving only the OTP calculation to the MQTT clients.

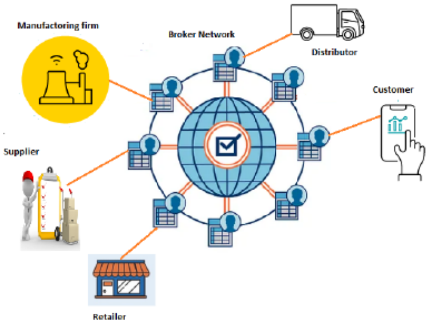


Fig. 1. MQTT architecture for supply chain management [4].

A. AI-Enhanced MQTT Architecture

As depicted in Fig. 2, the new architecture has introduced an AI component to continuously analyze the devices's behavior that are connected or in the process of connecting, to detect potential fraud. Once fraud is detected, a transaction is performed to invoke the smart contracts and apply the required security countermeasures. Before integrating the AI model into the operational MQTT architecture, it was first trained using test datasets simulating a real state of the MQTT traffic, and

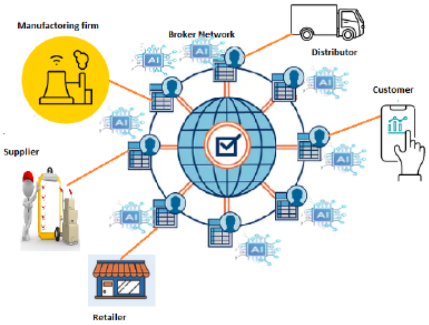


Fig. 2. AI-enhanced MQTT architecture.

then the model was tested. It is also important to note that the AI model will be permanently improved in order to integrate new potential attack types. The key steps of building and improving the AI model will be detailed in Section IV-A1.

1) AI model for device behavior analysis: The approach used to build and improve the AI model is based on hybrid learning, combining supervised learning to detect known attacks and unsupervised learning to discover new patterns of unknown attacks or other parameters for detecting malicious devices. The key stages of our approach are detailed below:

a) Stage 1: Defining the initial parameters using supervised learning
This phase aims to design an AI model that allows the prediction of malicious devices based on a set of parameters defined in Table II. The model is based on supervised learning, which uses labeled data for the model training to predict device Tbehavior subsequently. **Workflow 3** defines the steps to be followed during this phase, and is created using KANIME software.[1]
Workflow detail
- Dataset: The dataset used by the AI model in this phase either for training and testing is available via the link[2]; it simulates real-world network conditions during communication between an MQTT broker and eight sensors, and includes both normal and malicious activities. The behavior of devices is identified based on a set of relevant parameters, as detailed in Table II. The CSV Reader node is employed to load this data into KNIME.
- Data Pre-processing: This step consists of data preprocessing, which involves preparing the raw data so that it can be used effectively by the machine learning algorithms. This includes encoding categorical variables, and scaling numerical features [14].To do this, a **Label Encoding method** which assigns a unique integer to each category of a variable, and **Standard scaling** method which consists of centering the data around 0 with a variance of 1 are used, respectively. The **Category to Number** and **Normalizer** KNIME nodes are used for this purpose.
- Data Splitting: Using the Partitioning KNIME node, the dataset was divided into two categories: 70% for training and 30% for testing.
- Model selection: The model used in this article is the **Random Forest (FR)** classifier, as it provides the higher accuracy (91%) compared with other models k-nearest neighbors (KNN) (90%) and Decision Tree (85%) [15], and it is faster and more scalable for a huge database with many features compared to Support Vector Machine (SVM) which becomes very expensive in terms of computing time and memory for large databases [16]. Random Forest (RF) is an ensemblistic classifier that works by creating multiple decision trees. Each tree is built using a random subset of training samples and variables. It is based on Bootstrap Aggregating where each tree is trained on a different sample of data, drawn with a discount. Each decision tree in the forest makes its own prediction based on the input features. The final result of the RF classifier is determined by aggregating the predictions of all the individual trees, usually by majority vote for classification tasks or by average for regression tasks [17].
- Model training and Evaluation: In this phase, the model was trained and then tested using the training dataset (70%) and test dataset (30%), respectively. For this purpose, the **Random Forest Learner** and **Random Forest Predictor** KNIME nodes are used. The **Scorer** KNIME node is used for evaluation.

Interpretation and discussion of the results: To evaluate the model's ability to identify malicious behaviors, the following metrics were used:
- Accuracy: This metric measures the model's ability to correctly classify devices based on their behavior. In our case, as shown in Fig. 4, the model correctly classified 95.97% of the devices.
- Recall: This metric indicates the proportion of actual malicious behaviors correctly detected by the model. It is calculated using the following formula: $Recall = \frac{TP}{TP+FN} = 0.937$ TP (True Positive): The number of malicious behaviors correctly identified as such by the model. According to the confusion matrix (Fig. 4), this value is 46,492. FN (False Negative): The number of malicious behaviors that the model failed to detect. From the same confusion matrix, this value is 3,110. **Thus, the model successfully detected 93.7% of the malicious behaviors.**
- F1 Score: This metric evaluates the balance between the model's precision (ability to avoid misclassifications) and recall (ability to detect malicious behaviors). It is calculated using the formula:
$F1\ Score = 2 \times \frac{\text{Accuracy}\times\text{Recall}}{\text{Accuracy}+\text{Recall}} = 0.97$

These results indicate that the model offers strong overall performance, with an excellent compromise between the ability to detect malicious behavior and avoid misclassification of devices. Hence, upon completion of this phase, the model can be considered robust enough for deployment within the real-world MQTT architecture to detect all malicious behaviors it has been trained on.

b) Stage 2: Discovering new parameters and patterns for device behavior analysis using unsupervised learning
In this step the unsupervised learning algorithm will be incorporated into the real-world MQTT architecture to

[1] Knime, https://www.knime.com/
[2] MQTTset, https://www.kaggle.com/datasets/cnrieiit/mqttset

TABLE II. FRAUD DETECTION PARAMETERS OVERVIEW

Parameters	Description
IP source address	The device's IP address can help identify abnormal behavior, such as connecting from unusual regions or connecting with an IP address that belongs to the same address range as a device already detected malicious.
Geolocation	The device's location can be used to detect suspicious connections from unusual or unauthorized regions.
Timestamp	Timestamps can detect abnormal activities, such as connections occurring at unusual hours, inconsistent time intervals between connections, or instances where publish and subscribe messages are sent before establishing a connection.
Message size and format	An unusual size or format of MQTT messages can be used to identify malicious devices.
Messages Frequency	This parameter is used to detect abnormal activity and behavior, such as multiple simultaneous MQTT connections from source IP addresses within the same range, which may indicate that the device is part of a botnet network.
Session duration	This parameter can identify irregular behavior or activity. For instance, a device that frequently connects and disconnects may be flagged as suspicious.
Open ports	Open ports other than those used by the MQTT protocol or those used by standard applications can identify malicious devices.
Identifier format	A device could be classified as malicious if its identifier format deviates from the standard format.

adapt to new forms of fraudulent behavior by analyzing real-time traffic and identifying deviations from normal patterns. This approach enables us to discover new parameters that were not previously identified. However, due to the constraints of working within a real MQTT architecture, we will use NS-3 [18] network simulation environment in order to generate different types of traffic and behaviors corresponding to both legitimate and malicious activities. It's also important to note that no security measures are applied to the anomalies detected during this phase. **Worfflow 5** defines the steps to be followed during this phase, and it is created using KANIME software (Fig. 3, 5).

Worflow detail

- Data preparation: After running the simulation script in the NS-3 simulator, the data will be logged into "simulation_trace.csv".This file will be loaded into KNIME using the **CSV reader** node.
- Data Pre-processing: This step pre-processes the data to make it suitable for the Isolation Forest algorithm, the unsupervised learning algorithm used in this phase. This includes handling missing values, encoding categorical data, and scaling data. The KNIME nodes used for this purpose are: **Missing value** to handle missing values by replacing them with a mean value, **One to many** to encode categorical data, and **Normalizer** to scale the data.
- Application of Isolation Forest: In this step, the Isolation Forest algorithm was applied to identify potential anomalies in the dataset. The isolation forest calculates an anomaly score for each observation in the dataset. This score provides a measure of the normality of each observation relative to the entire dataset. To calculate this score, the algorithm isolates the dataset in question in a recursive manner: it chooses a variable at random and sets a random cut-off point, then evaluates whether this isolates a particular observation [19]. The KNIME node used for this purpose is the **Isolation Forest** node.
- Anomalies analysis: Fig. 6, 7 depict the most relevant parameters that can be added to the initial parameters for devices behavior analysis. As shown in Fig. 6, for all parameters, normal instances are centered around the mean (0) with low standard deviations, while abnormal instances deviate significantly from the mean with higher standard deviations. The above parameters can be summarized as follows:
  Device physical features: This parameter includes a set of device characteristics namely computing power (GHz), storage capacity (GB), and memory (GB). As shown in Fig. 6, the abnormal instances have significantly higher CPU, storage, and memory values than the normal ones. Therefore, we can say that this parameter is relevant for detecting malicious devices since the sensors are generally equipped with low computing power, storage capacity, and memory.
  Device resources consumption: This parameter measures the rate of resource consumption, more specifically the percentage of CPU utilization (%) and power consumption (W). As shown in Fig. 6, the abnormal instances tend to consume more energy and processing power, and this is likely due to the deployment of heavy malicious applications or abnormal processing. Hence this parameter is considered as relevant for identifying malicious devices.
  Installed software: This parameter is used to identify the number of applications installed on the devices. As can be seen in Fig. 6, abnormal instances contain a relatively large number of applications compared with normal instances, so this parameter can also be used to identify malicious devices, since the applications installed in sensors are generally limited.
  Firmware type and version: This parameter is used to identify the type and version of firmware installed on the devices. To facilitate the interpretation of the results for this parameter, we have converted the numerical values used by the model into two categories (Fig. 7): **Standard**, which includes known firmware with recent versions, and **unkown**, which contains unknown firmware, obsolete or test versions. This parameter is also relevant for identifying malicious devices.

Conclusion: The relevant parameters we identified during this phase are **device physical features, device resource consumption, and firmware type and version**. These parameters will be added to the initial parameters to improve the accuracy of the model and help better identify malicious devices.

c) Stage 3: Model enhancements:
The new parameters identified by the unsupervised learning algorithm will be fed back into the supervised model to refine detection criteria and improve model accuracy.

2) The communication phases: The communication between the brokers network and clients (Publisher/subscriber) occurs in three main steps:

- Registration phase: In this phase, each device must be registered in the blockchain by a trusted administrator.

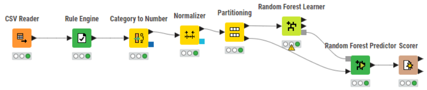
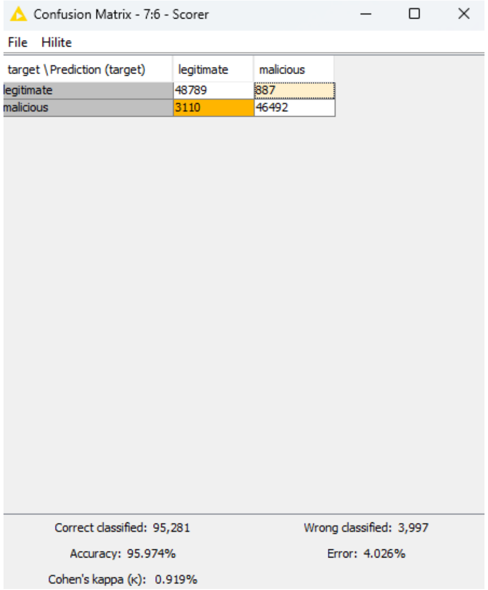


Fig. 3. AI-model workflow (phase 1).



target \ Prediction (target)	legitimate	malicious
legitimate	48789	887
malicious	3110	46492

Correct classified: 95,281

Wrong classified: 3,997

Accuracy: 95.974%

Error: 4.026%

Cohen's kappa (κ): 0.919%

Fig. 4. Confusion matrix.

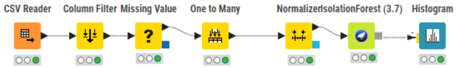


Fig. 5. AI-model workflow (phase 2).

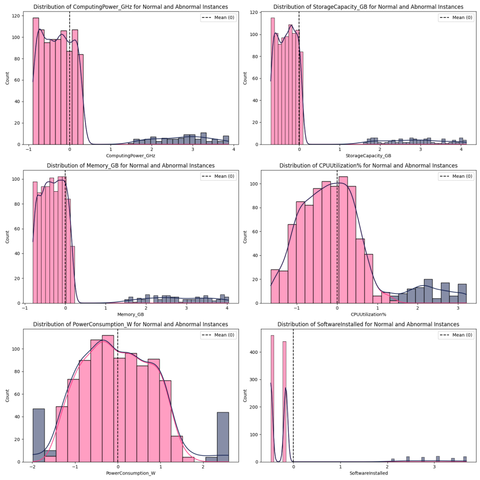


Fig. 6. Relevant parameters distribution for normal and abnormal instances.

The latter assigns publication and subscription rights to specific topics and, in return, he retrieves the required keys for authentication and message encryption and communicates them to the devices in out-of-band mode. When an administrator registers a device, its reputation is set to true by default. The exchange details are illustrated in sequence Fig. 8 and the functioning of the smart contract responsible for this phase is described in the activity Fig. 9.

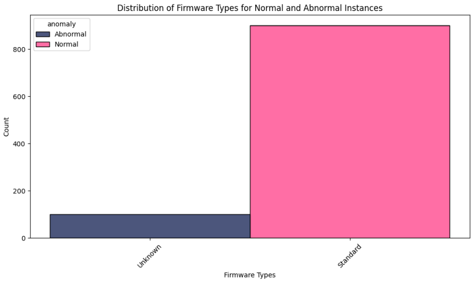


Fig. 7. Firmware type distribution for normal and abnormal instances.

- Connection phase: As depicted in the sequence Fig. 10, when the brokers network receives a connection request, it verifies the packet number, the device registration, and the reputation. If the device is not registered, the connection will be denied and the device will be added to the blockchain with a reputation set to False. Then, if the device's reputation is false, the connection to the brokers network is denied; otherwise, a challenge is sent to a device for OTP calculation. The device calculates the OTP and sends the hashed OTP. If it is correct, the device is connected to brokers network else, the connection is denied and the reputation is set to false . Simultaneously, the AI model constantly analyzes device behavior. Once it detects malicious behavior, it invokes the smart contract to interrupt the connection process and change the device's reputation to false. The functioning of the smart contract responsible for this phase is described in the activity Fig. 11.
- Publishing phase: As depicted in sequence Fig. 12, in this phase, once a device publishes a message to brokers network containing a topic name, and encrypted data using the secret topic key, the smart contract checks its rights to publish to that topic as well as the data integrity and then it notifies all the subscribers to that topic. If the client doesn't have the right to publish in this topic, the connection is automatically interrupted and the reputation is changed to false. During this phase, if the AI model detects fraud, it invokes the smart contract to interrupt the connection and change the device's reputation to false. The functioning of the smart contract responsible for this phase is described in the activity Fig. 13.

V. Attack Scenarios Analysis

For the test simulation, we apply the proposed architecture in Supply Chain Management as an IOT critical application and perform the tests in the Remix environment.[3] Indeed, the proposed solution fully fit the requirement of the supply chain management since it requires the intervention of multiple independent entities such as suppliers, manufacturers, distributors, retailers, and end customers. Each entity has the right to publish and subscribe to specific topics. These rights

[3] Remix IDE, https://remix.ethereum.org/

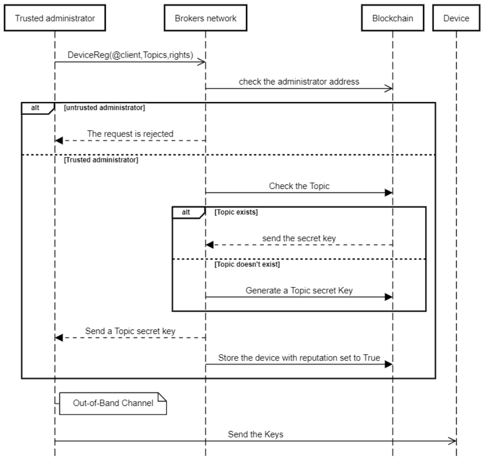


Fig. 8. Registration phase sequence diagram.

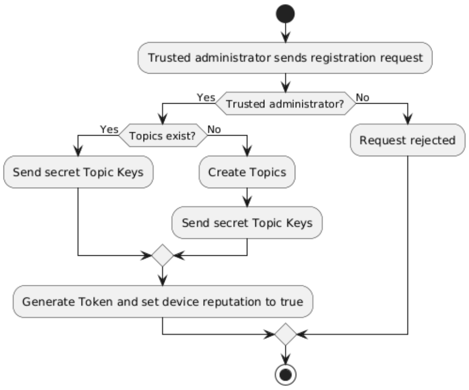


Fig. 9. Registration phase smart contract logic.

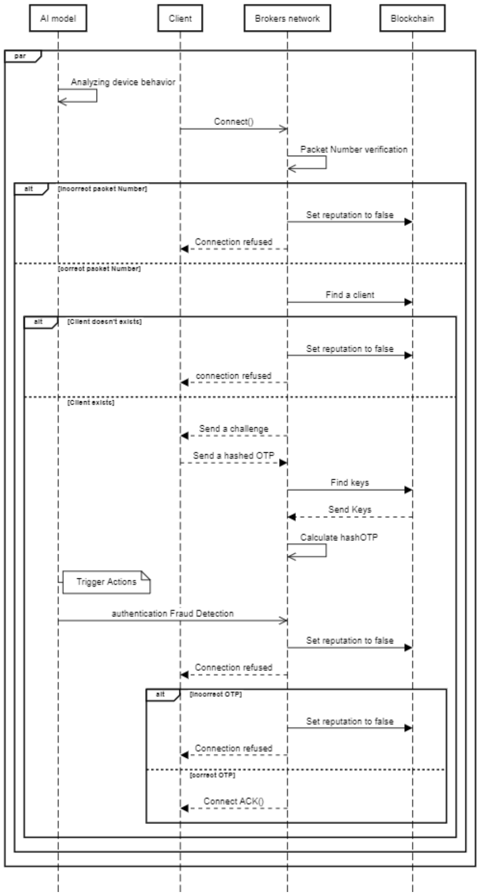


Fig. 10. Connection phase sequence diagram.

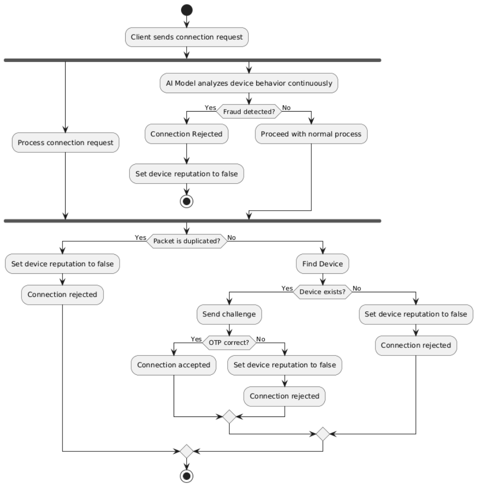

Client sends connection request
AI Model analyzes device behavior continuously
Yes
Fraud detected?
No
Process connection request
Connection Rejected
Proceed with normal process
Set device reputation to false
Yes
Packet is duplicated?
No
Set device reputation to false
Find Device
Connection rejected
Yes
Device exists?
No
Send challenge
Set device reputation to false
Yes
OTP correct?
No
Connection rejected
Connection accepted
Set device reputation to false
Connection rejected

Fig. 11. Connection phase smart contract logic.

are granted by the administrator of each entity during the registration phase.

A. Test Cases

- Scenario 1: Registration nominal scenario
  As an example, the Manufacturer's administrator registers the temperature sensor for the concerned entity. As shown in Fig. 14, they assign write-only permissions to the topic "Man_smart_sensor/temperature". Upon registration, the sensor's reputation is automatically set to "true" by default.
- Scenario 2:Attempted Connection of Unregistered Device
  As depicted in Fig. 15, when an unregistered device is connected to the brokers network, the connection is refused and the device is registered in the blockchain with the reputation equal to false.
- Scenario 3: Device connection with false reputation
  As depicted in Fig. 16, when a device with a false reputation attempts to connect to brokers network, the connection is automatically denied.
- Scenario 4: Submission of incorrect OTP
  - Manufacturer's temperature sensor sends a connection request to the brokers network.
  - After checking the packet number, device registration, and reputation, the brokers network send the challenge to the Manufacturer's temperature sensor for OTP calculation.
  - Manufacturer's temperature sensor sends an erroneous OTP.
  - As depicted in Fig. 17, the connection is automatically refused and the reputation is set to false.
- Scenario 5: Unauthorized Publication

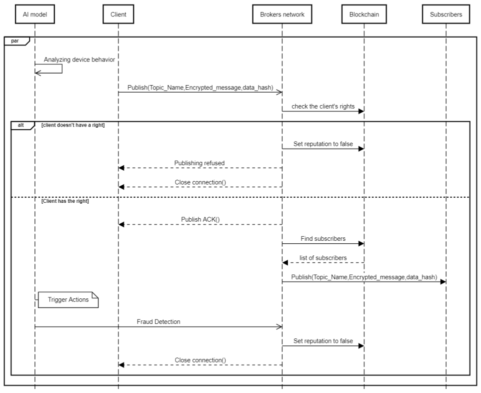


Fig. 12. Publishing phase sequence diagram.

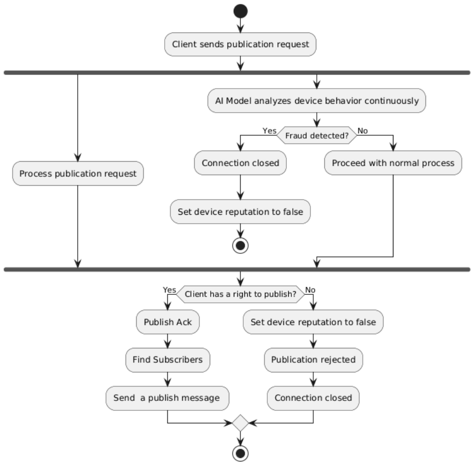

Client sends publication request
AI Model analyzes device behavior continuously
Yes
Fraud detected?
No
Connection closed
Proceed with normal process
Process publication request
Set device reputation to false
Yes
Client has a right to publish?
No
Publish Ack
Set device reputation to false
Find Subscribers
Publication rejected
Send a publish message
Connection closed

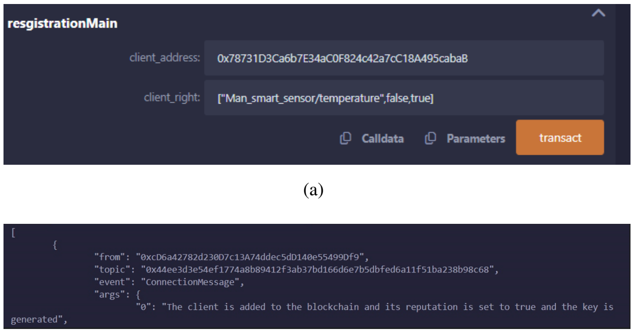
Fig. 13. Publishing phase smart contract logic.

- ○ As an example, the distributor's administrator has registered the GPS device, allocating the rights described in Table III to receive updates or instructions regarding its configuration.
- ○ As shown in Fig. 18, when the distributor's GPS is attempting to publish in an unauthorized topic **Smart_Sensorupdateconfiguration**, the connection is interrupted and its reputation is set to false.

- Scenario 6: Detection of Known Authentication Fraud
  - ○ During the connection process of a customer's end device, the AI model classified the device as malicious due to a malformed connection packet.
  - ○ The AI model invokes the smart contract to deny the connection and set the device's reputation to false **Fig. 19**.
- Scenario 7: Detection of abnormal known behavior
  The AI model detected a flood attack from an already connected customer's end device. As depicted in Fig. 20, it triggered the smart contract to interrupt the connection and set the device's reputation to "false".

B. Discussing Test Results

In Section V-A, we have focused on simulating attacks related to the new concepts introduced in this article, which



resgistrationMain

client_address: 0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB

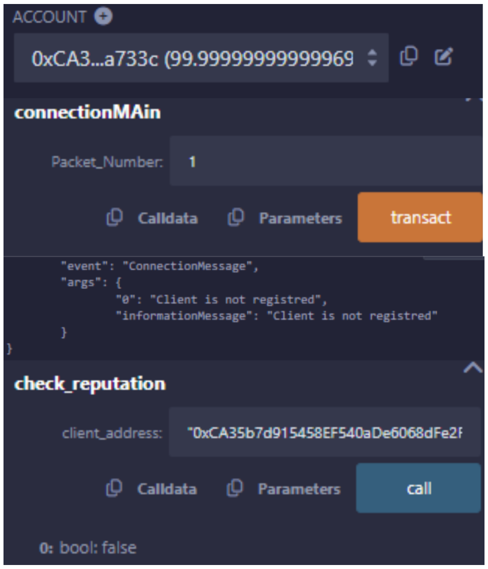
client_right: ["Man_smart_sensor/temperature",false,true]

(a)

```
[
	{
		"from": "0xcD6a42782d230D7c13A74ddec5dD140e55499Df9",
		"topic": "0x44ee3d3e54ef1774a8b89412f3ab37bd166d6e7b5dbfed6a11f51ba238b98c68",
		"event": "ConnectionMessage",
		"args": {
			"0": "The client is added to the blockchain and its reputation is set to true and the key is
generated",
```

(b)

Fig. 14. Registration nominal scenario.



ACCOUNT

0xCA3...a733c (99.9999999999996

connectionMAin

Packet_Number: 1

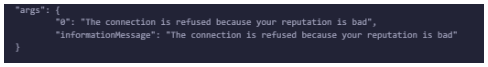
```
		"event": "ConnectionMessage",
		"args": {
			"0": "Client is not registred",
			"informationMessage": "Client is not registred"
		}
}
```

check_reputation

client_address: "0xCA35b7d915458EF540aDe6068dFe2F

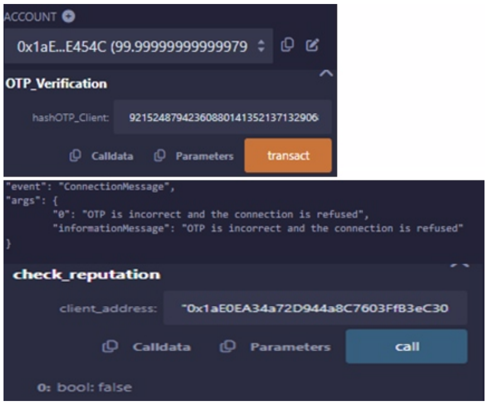
0: bool: false

Fig. 15. Connection of unregistered device.



```
"args": {
	"0": "The connection is refused because your reputation is bad",
	"informationMessage": "The connection is refused because your reputation is bad"
}
```

Fig. 16. Connection denied due to false reputation.



ACCOUNT

0x1aE...E454C (99.999999999999979

OTP_Verification

hashOTP_Client: 92152487942360880141352137132906

```
"event": "ConnectionMessage",
"args": {
	"0": "OTP is incorrect and the connection is refused",
	"informationMessage": "OTP is incorrect and the connection is refused"
}
```

check_reputation

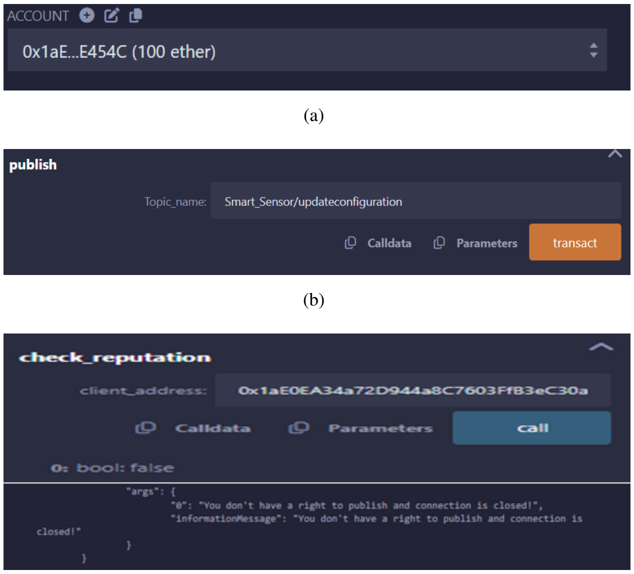
client_address: "0x1aE0EA34a72D944a8C7603FfB3eC30

0: bool: false

Fig. 17. Submission of incorrect OTP.

TABLE III. SMART SENSOR REGISTRATION INFORMATION

Device address	”0x1aE0EA34a72D944a8C7603FfB3eC30a6669E454C”
Topic Name	GPS/updateconfiguration
Read right	True
Write right	False



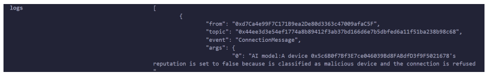
```
                "args": {
                        "0": "You don't have a right to publish and connection is closed!",
                        "informationMessage": "You don't have a right to publish and connection is
closed!"
                }
        }
```

(a)

(b)

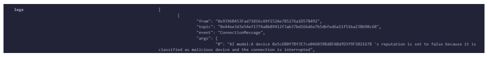
(c)

Fig. 18. Unauthorized publication.



```
logs                              [
                                        {
                                                "from": "0xd7Ca4e99F7C171B9ea2De80d3363c47009afaC5F",
                                                "topic": "0x44ee3d3e54ef1774a8b89412f3ab37bd166d6e7b5dbfed6a11f51ba238b98c68",
                                                "event": "ConnectionMessage",
                                                "args": {
                                                        "0": "AI model:A device 0x5c6B0f7Bf3E7ce046039Bd8FABdfD3f9F5021678's
reputation is set to false because is classified as malicious device and the connection is refused
```

Fig. 19. Authentication fraud detection.



```
logs                              [
                                        {
                                                "from": "0x9396B453Fad71816cA9f152Ae785276a1D578492",
                                                "topic": "0x44ee3d3e54ef1774a8b89412f3ab37bd166d6e7b5dbfed6a11f51ba238b98c68",
                                                "event": "ConnectionMessage",
                                                "args": {
                                                        "0": "AI model:A device 0x5c6B0f7Bf3E7ce046039Bd8FABdfD3f9F5021678 's reputation is set to false because it is
classified as malicious device and the connection is interrupted",
```

Fig. 20. Abnormal behavior detection.

follow on from the attack tests carried out in the previous work [4]. Attack simulation tests focus on two main areas: on the one hand, the interaction between the artificial intelligence (AI) model and the smart contracts to apply the appropriate security measures once malicious behavior has been detected. Indeed, as illustrated in **scenario 6**, the AI model analyzes the device’s behavior during connection process and, as a result, denies connection to the brokers network if malicious behavior is detected. In **scenario 7**, malicious behavior is detected while the device is already connected to the brokers network, triggering hence a transaction to interrupt the connection. On the other hand, these tests also monitor the device’s reputation status, which must automatically change if malicious behavior is detected. In fact, according to scenario 1, by default, the state of the reputation when the device is registered is good **(true)**. Once malicious behavior is detected, the reputation changes to bad **(false)**. This is illustrated in the following scenarios: connection of an unregistered device **(scenario 2)**, sending a wrong OTP **(scenario 4)**, unauthorized publication **(scenario 5)**, and detecting abnormal behavior using the AI model **(scenarios 6 and 7)**. Additionally, the main aim of the device reputation system is to prevent devices with a bad reputation from rejoining the network, as demonstrated in **(scenario 3)**.

VI. CONCLUSION

Our research works aim to propose an MQTT architecture that meets the security requirements of critical IoT applications, is resilient to current attacks, and is adaptable to potential future attacks while taking into account the constraints of the IOT environment. To this end, the proposed solution in this paper aims to improve the MQTT architecture proposed in our previous work [4] by adding an additional layer of security based on artificial intelligence. Combining the advantages of blockchain and AI technologies enables attack detection by analyzing the behavior of devices connected or being connected to the broker network using an AI hybrid model, and then automatically applying appropriate security measures using smart contracts. The solution has also introduced the concept of device reputation to prevent malicious devices from rejoining the network. The creation of the AI model involved three essential phases: The first step was to train the model on a set of known malicious behaviors using the Random Forest supervised learning algorithm. Once the model was trained, it was integrated into the MQTT architecture, where the appropriate security measures were implemented through smart contracts. Simultaneously, the Isolation Forest unsupervised learning algorithm was added to the model in monitoring mode, in order to discover new attack patterns and identify new parameters for the detection of malicious devices. The attack patterns identified in phase 2 will be reintegrated into the basic model to improve the model’s accuracy and performance. To fit the constraint environment requirements, all the resource-intensive operations are managed on the brokers network side, while the only operation executed on the device side is the OTP calculation. For attack simulation tests, the architecture was applied to supply chain management, and smart contracts were implemented in the Remix environment. The test results showed the architecture’s resistance to different types of attacks. In our future work, we will continue to improve the security of the MQTT protocol in constrained environments by deploying our architecture in real-world situations. This will allow us to expose the architecture to real attacks, which will refine the AI model and strengthen the architecture’s resilience against more complex and sophisticated types of attacks. In addition, we will evaluate the performance of the MQTT protocol by measuring key indicators such as latency, energy consumption, and bandwidth utilization.

REFERENCES

[1] M. Tauseef, M. R. Kounte, A. H. Nalband, and M. R. Ahmed, “Exploring the joint potential of blockchain and ai for securing internet

of things," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023.

[2] "Internet of things - worldwide — statista market forecast," https://www.statista.com/outlook/tmo/internet-of-things/worldwide,date2024-08-20.

[3] T. P. HT, T. N. DP *et al.*, "Developing a patient-centric healthcare iot platform with blockchain and smart contract data management." *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 4, 2024.

[4] R. AKNIN and Y. Bentaleb, "Enhanced mqtt architecture for smart supply chain," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, 2023.

[5] F. Buccafurri, V. De Angelis, and R. Nardone, "Securing mqtt by blockchain-based otp authentication," *Sensors*, vol. 20, no. 7, p. 2002, 2020.

[6] P. Akshatha and S. D. Kumar, "Mqtt and blockchain sharding: An approach to user-controlled data access with improved security and efficiency," *Blockchain: Research and Applications*, vol. 4, no. 4, p. 100158, 2023.

[7] T.-C. Hsu and H.-S. Lu, "Designing a secure and scalable service model using blockchain and mqtt for iot devices," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 645–653.

[8] R. Aknin and Y. Bentaleb, "Securing mqtt architecture using a blockchain," in *Advances in Information, Communication and Cybersecurity: Proceedings of ICI2C'21*. Springer, 2022, pp. 568–578.

[9] R. Alasmari and A. A. Alhogail, "Protecting smart-home iot devices from mqtt attacks: An empirical study of ml-based ids," *IEEE Access*, vol. 12, pp. 25 993–26 004, 2024.

[10] A. Alzahrani and T. H. Aldhyani, "Artificial intelligence algorithms for detecting and classifying mqtt protocol internet of things attacks," *Electronics*, vol. 11, no. 22, p. 3837, 2022.

[11] 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.

[12] M. A. Jabraeil Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, F. Norouzi, M. A. Jabraeil Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, "Iot architecture," *Towards the Internet of Things: Architectures, Security, and Applications*, pp. 9–31, 2020.

[13] A. Laftimi, H. El Makhtoum, R. Aknin, and Y. Bentaleb, "Ai-based intelligent blockchain for the authentication of the metering system," in *2022 IEEE 3rd International conference on electronics, control, optimization and computer science (ICECOCS)*. IEEE, 2022, pp. 1–6.

[14] "Qu'est-ce la data preparation en machine learning ?" https://www.free-work.com/fr/tech-it/blog/actualites-informatiques/impact-de-la-data-preparation-en-machine-learning,date2024-08-10.

[15] P. Dinesh, A. Vickram, and P. Kalyanasundaram, "Medical image prediction for diagnosis of breast cancer disease comparing the machine learning algorithms: Svm, knn, logistic regression, random forest and decision tree to measure accuracy," in *AIP Conference Proceedings*, vol. 2853, no. 1. AIP Publishing, 2024.

[16] C. Avcı, M. Budak, N. Yağmur, and F. Balçık, "Comparison between random forest and support vector machine algorithms for lulc classification," *International Journal of Engineering and Geosciences*, vol. 8, no. 1, pp. 1–10, 2023.

[17] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS journal of photogrammetry and remote sensing*, vol. 114, pp. 24–31, 2016.

[18] "ns-3," https://www.nsnam.org/,date2024-08-10.

[19] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE transactions on knowledge and data engineering*, vol. 33, no. 4, pp. 1479–1489, 2019.