Machine Learning-Based Detection and Mitigation of DDoS Attacks in MQTT Communication for IoT Systems.

W.T.T. PEIRIS
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri lanka
it21096334@my.sliit.lk

Abstract— **The applicability of machine learning in the detection and mitigation of DDoS attacks within the IoT environments that employ the MQTT communication protocol is discussed in this research paper. Various security measures available in current research, with real-time stronger solutions, have been reviewed and pointed out for optimization at MQTT. This provides an insight into the strengths and limitations of different machine learning approaches with a focus on increasing security in IoT devices. The elicited knowledge is intended to have an impact on the development of intrusion detection systems that would be more counterbalancing against special challenges brought about by IoT due to its scale, dynamism, and diversity.**

Keywords—Decision Tree, Distributed Denial of Service (DDOS), Internet of Things (IoT), machine learning (ML), Message Queuing Telemetry Transport (MQTT)

I. INTRODUCTION

The Internet of Things has significantly changed the world of The IoT has fostered huge interconnections in the device from most industries, including health, manufacturing, smart cities, and agriculture. Devices have huge interconnections that allow the ability of sharing data smoothly, enhancing automation, operational efficiency, and innovation in many applications. Rapid adoption of IoT shall foster industries with real-time insight, smoothening processes, innovating new business models.

The rapid growth in IoT, increases the number of connected devices as well as the critical nature of data transferred by them, further indicating the dire need for security measures. Since most of the IoT devices move in a resource-constrained and distributed environment, it is quite easy to be hit by cyber-attacks. Therefore, security in the IoT is not only a technical issue but also a fundamental requirement from the viewpoint of data integrity, confidentiality, and availability, as well as reliable operation of critical infrastructures.

There is a high demand for robust security in IoT, as severe consequences flow from security breaches. These include, but are not limited to, data theft, sensitive information compromise, and privacy violation-all of which erode public trust. Additionally, disruptions to vital services related to health, energy, or transportation are invariably likely to pose a huge risk to societal well-being and economic stability. The very worst breaches cause physical harm-something that is especially possible in IoT deployments that run critical infrastructure, such as industrial or medical. Given the increasing integration of IoT devices into daily life and into critical systems, the need for policies that ensure their comprehensive security is increasingly urgent. That is highly important not only for securing the needed data and providing privacy protection but also for keeping reliability and safety of the infrastructure supporting modern society. Accordingly, securing IoT systems is the precondition for maintaining benefits accruable from this fast-growing technology [1].

IoT networks are transformative in nature, with a huge number of devices strung together normally having limited available security measures, these are turning out pretty susceptible to all kinds of cyberattacks. These include, among others, distributed denial of service, data breaches, man-in-the-middle attacks that considerably inhibit operations in IoT. The most deployed communication protocol in IoT is MQTT which stands for Message Queuing Telemetry Transport, and is a lightweight and efficient protocol. Because of the least complexity in design, MQTT is quite an alluring target for attackers. This paper proposes how to detect and mitigate DDoS attacks using machine learning methods in MQTT communications for security and reliability in IoT systems. [2].

II. literature review

The sudden growth in the Internet of Things has transformed security in IoT networks into a focus area. On one hand, greater integration of IoT devices across various sectors has opened up new avenues of innovation and efficiency in fields such as healthcare, industry, and smart cities. On the other hand, however, it has also exposed tremendous vulnerabilities and challenges, more related to network security. This literature review covers unique security challenges of IoT networks; the focus will be on vulnerabilities within the systems and current developments that could offer improvements, especially in the use of machine learning techniques in finding and mitigating cyber-attacks..

A. Security Challenges in IoT Systems

IoT systems are highly diverse, with devices varying in computational power, hardware design, and communication protocols. This variation introduces significant security vulnerabilities. Most IoT devices lack adequate computational resources, preventing the implementation of robust security measures like advanced encryption or intrusion detection. Their low cost and efficiency prioritize functionality over security, leaving many devices vulnerable to attacks. The heterogeneity of devices further complicates security, as solutions effective for one device may not work for another, leading to blind spots. Additionally, the lack of standardized security protocols and interoperability across devices exacerbates the risks, making IoT systems easy targets for cyberattacks [3].

The threat of cyber-attacks on the devices of Industrial IoT and SCADA systems is very much looming over all critical infrastructures such as energy, water, and transportation due to their increasingly sophisticating nature and rising frequency. While these kinds of systems provide vital functionality for monitoring, control, and automation, most of the weak security measures deployed by such systems turn them into an easy target for cybercriminals. The threats include DDoS attacks, ransomware, data sniffing, and APTs that may disrupt operations, result in theft of sensitive data, or cause physical damage [4] [5]. DDoS attacks can forcibly bring down industrial IoT systems through network overloads, resulting in the shutdown of production or failure in power distribution. Ransomware attacks, which encrypt critical information and demand a ransom for decryption, put the system at risk of failure if the ransom is not paid. The Stuxnet worm that attacked Iran's nuclear facilities was one of the most notorious cyber-attacks to hit an industrial IoT system. It tampered with the SCADA systems, causing centrifuges to malfunction and self-destruct while seeming normal to the human operators. This attack was a sobering development in a world where it was how IoT and SCADA systems can be used as weapons that create physical damage and significant disruptions to critical infrastructures [6]. An example is the cyberattack in 2015 against the Ukrainian power grid; in this case, actors that had penetrated its SCADA systems and eventually caused a large blackout that affected over 230,000 people [7]. The Stuxnet attack exemplifies the vulnerabilities of industrial IoT systems, particularly in the energy sector, where breaches can have severe consequences and exploit connected systems for physical and economic damage. [8].

Specifically, pacemakers, insulin pumps, and connected imaging systems are some of the IoT devices in the healthcare sector that could easily be targeted through cyberattacks [9] [10]. he 2017 WannaCry ransomware attack seriously affected healthcare facilities, plunging medical devices into going offline and disrupting the most vital treatments of patients. Researchers have remarked that the IoT devices are vulnerable due to outdated software and weak security configuration [11] [12]. These incidents put a milestone on the security needs of IoT and SCADA systems.

B. *DDoS Attacks in IoT*

DDoS attacks are the most frequent but threatening type of cyber-attack that aims at making the target network or service unavailable for legitimate users through overwhelming it with superfluous traffic in very excessive amounts. In IoT networks, the large number of interconnected devices is utilized for these attacks. [13]. Most of the devices deployed in any IoT application are characterized with limited computation capabilities and low security, making it easy for an attacker to compromise them easily. The same attacker could then build a botnet of such devices and launch a traffic storm against any target. The target network infrastructure gets overwhelmed with such amount of traffic, which leads to massive service disruption. Devices on IoT networks are diverse, from tiny home appliances to sensors within industries, which increases their vulnerability toward DDoS attacks. This means that just a small number of compromised devices can lead to huge attacks that have financial and operational impacts, especially in critical sectors like healthcare and energy. DDoS also triggers cascading failures in interconnected systems. Mitigation becomes tough because most IoT devices are diverse in nature, as also due to the lacking uniform security protocols [14]. The potential damage from DDoS attacks has grown with the integration of IoT devices into essential services and critical infrastructures, and much more effective security for protecting these networks against such threats has become urgent [15].

These IoT systems can be further differentiated into: volumetric, protocol, and application-layer attacks. In volumetric attacks, a congestion in the network occurs because of unusually high traffic; the bandwidth is consumed, and so the service becomes disrupted. The protocol variety takes advantage of the exhaustion of server resources, usually targeting TCP/IP protocols. In the application-layer kind of attacks, there is the sending of a high volume of legitimate requests for an application with the intent to flood it and render its functions inoperable. IoT systems can lead to critical service crippling, disrupted communications between devices, and wide-scale operational failures, especially in those applications relying on real-time data and control. [16].

C. *MQTT Protocol and Its Security Concerns*

IoT architecture usually contains a few layers, such as perception, normally done by sensors and devices; the network layer, which involves communication protocols; and the application layer, where the processing and analysis of data take place for end-user applications (Figure 1). The application layer is very important since it interfaces the user with provided services like monitoring, control, and analytics [17]. In IoT systems, several protocols may be utilized for data exchange, including HTTP, CoAP, Zigbee, Bluetooth Low Energy, and MQTT, their choice depending upon the requirements of a particular environment [18].

MQTT is an application-layer protocol in general use for IoT communication because of its efficiency, making it lightweight enough for low processing power and bandwidth devices. It enables real-time communication by allowing devices to publish data to brokers for distribution to subscribers (Figure 2) [19], it also enables several IoT applications, from simple smart home systems to industrial automation to autonomous vehicles [20]. Although MQTT offers a number of benefits, it also involves significant security risks, since it does not inherently use encryption and authentication mechanisms are weak. Without external encryption, such as TLS/SSL, data over MQTT can easily be sniffed and read. Besides, usually, the Authentication in MQTT is done with simple username-password authentication, which is easily broken, particularly in IoT contexts where default or weak credentials are widely used [21].

Architecture-wise, MQTT is highly susceptible to DDoS attacks. An attacker can connect to the broker multiple times or send multiple publish-subscribe messages to over-resource the broker and bring down the whole IoT network [22]. Additionally, the vulnerabilities of MQTT are further exposed without the inclusion of inherent encryption and with mere username-password authentication, hence putting data integrity, confidentiality, and availability at risk-especially in critical sectors such as industrial and healthcare systems. Security features that have to be considered in order to ensure security involve the implementation of TLS/SSL encryption, strong authentication protocols such as OAuth, rate limiting, and anomaly detection mechanisms in IoT systems using MQTT [21].

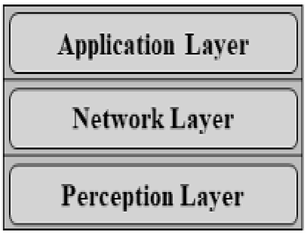


Figure 1. IOT Architecture

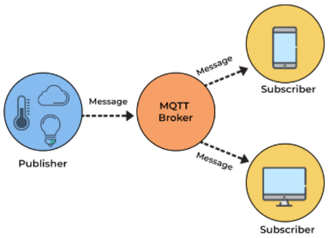


Figure 2. MQTT Process

D. *Existing Security Measures for IoT and MQTT*

This makes securing IoT-based networks, while using the MQTT protocol, tricky, since IoT devices are bound to limitations on computation power, memory, and energy. Traditional approaches to security, such as encryption, authentication, and firewalls, are severely limited in efficiency in the context of IoT. Specialized approaches are thus in order when it comes to protecting IoT networks from DDoS and other similar types of attacks.

Encryption through TLS in MQTT does indeed ensure the confidentiality of data in transit and protects against man-in-the-middle attacks. Usually, TLS is computing-intensive, and IoT devices have very minimal resources. TLS also doesn't provide solutions to weak configurations of devices or poor authentication [23].

The default authentication in MQTT is based on a username-password and, furthermore, is weak. Advanced techniques for improving authentication in MQTT include OAuth 2.0 and Certificate-based authentication. OAuth 2.0, which does not require one to expose his credentials, is secure and flexible. Regarding its part, Certificate-based authentication is able to verify the device's identity. However, managing all the certificates can be quite complex and very resource-intensive, especially in the case of extensive IoT deployments [24] [25].

Firewalls in IoT networks manage numerous types of access, including access denial to MQTT brokers and rate limitations for preventing DDoS attacks. Traditional firewalls are defeated by the dynamic nature of IoT, where devices frequently join or leave the network, and do not protect against attacks from compromised IoT devices within the network [26] [27].

Anomaly detection systems can identify unusual behavior from MQTT clients, such as excessive connection requests or messages. By detecting these patterns, these systems help prevent DDoS attacks by blocking or throttling malicious clients, thus increasing broker resilience [28].

Network segmentation provides another layer of security related to IoT, which can keep DDoS attacks within a particular network segment. Each such segment can have its own firewalls and access controls that prevent proliferation across other segments, reducing the attack surface on the whole [29].

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are being adapted for IoT environments to monitor network traffic for malware or other attacks. In the case of MQTT networks, IDS and IPS can detect some types of attack-like authentication-related issues or flooding the broker with connection requests-to prevent or mitigate the threat [30].

All security solutions based on the cloud provide an extendable way of securing MQTT communications by offloading security processes to the cloud. In this respect, it supports advanced security measures such as DDoS protection, which absorbs large-scale attacks before they reach your MQTT broker and ensures that your IoT network remains operational even during attacks [31].

While these are important in IoT and MQTT, encryption, authentication, and firewalls are not sufficient on their own. Only an integrated security solution will constitute a robust approach, part of which includes specialized solutions in the form of rate limiting, anomaly detection, network segmentation, IDS/IPS, and cloud-based services that are integral to any strategy for the efficient protection of evolving IoT network threats.

E. *Machine Learning in Cybersecurity*

Machine learning has been one of the vital technologies that have been used in cybersecurity for IoT systems. Traditional rule-based security, due to the sophistication of cyberattacks, has faced barriers in finding new and evolving threats. ML does analyze large datasets for patterns to provide better security to networks and devices by offering advanced threat detection capabilities.

ML also contributes to cybersecurity in the field of threat detection through signature recognition and profiling of abnormal behavior. Supervised learning, in particular, applies when the model needs training on how to classify data as either benign or malicious. This is particularly helpful in those cases when the normal and abnormal kinds of behaviors are well defined, for example, in the case of a phishing or malware attack [32].

Unsupervised learning becomes useful when labeled data is limited or the nature of the attacks is unknown. It does not rely on any predefined labels but instead discovers patterns within the data. This technique goes well in anomaly detection for IoT security, as it flags variations from normal behavior, such as sudden spikes in network traffic or some unpredictable device behavior, which may indicate an attack [33].

Reinforcement learning in cybersecurity continues to gain more and more prominence, especially in developing adaptive systems that can respond in real-time to newly emerging threats. Taking up such an approach means the agent interacts with the environment, tuning security measures, such as firewall rules and access controls, to evolving threats for more resilient and adaptive defenses, especially within IoT networks. [34].

ML automates responses like quarantine compromised devices or applying security patches, something quite necessary in complex IoT systems, wherein many cases human intervention is simply not feasible. This fortifies IoT security by detecting and responding to such events in real time [35]. This will require full understanding of how mitigation strategies secure IoT systems and MQTT communications against DDoS attacks using machine learning; therefore, many machine learning approaches and how they contribute to intrusion detection have to be discussed.

Neural Networks, especially the deep learning models, hold immense potential in conducting anomaly detection across IoT systems. Convolutional Neural Networks (CNNs) can review the traffic data to discover abnormal traffic patterns [36]. Recurrent Neural Networks (RNNs) help in the identification of temporal patterns in data, proving helpful in the detection of sequential attacks, such as DDoS [37].

Support Vector Machines (SVMs) are a popular supervised learning algorithm that works effectively for binary classification problems like those needed for discriminating whether a traffic is normal or malicious. The features analyzed by an SVM in DDoS detection involve packet size and incoming request rates to classify traffic either as benign or harmful [38].

Decision trees are widely employed on IoT network anomaly detection based on their simplicity and interpretability. They classify the network traffic with features like IP addresses and packet sizes with which to track down the patterns of malicious traffic, such as DDoS attacks. Decision trees become really worth their weight in gold in situations that call for an explanation of decisions [39]. The model was therefore trained using datasets labeled to trace back malicious traffic patterns, for example, DDoS attacks. A clear advantage is shown by decision trees in scenarios where explainability is paramount by giving an apparent reason for each decision made [40].

Random Forests perform ensemble learning by collecting the output of several decision trees, hence reducing overfitting and enhancing generalization. It is effective in detecting various types of attacks, such as DDoS, Low-Rate DoS LDOS, and Man-in-the-Middle. Random Forests are robust against noise and suitable for IoT network complexity [41].

K-Nearest Neighbors (KNN) is a quite simple, yet effective machine learning technique applied to anomaly detection in IoT networks. KNN works by comparing a new data point with its K top neighbors according to the training set. In the case where most of the neighbors majorly belong to a certain class, for example, normal or malicious, a new data point is classified accordingly. In particular, KNN has strong applications in detecting anomaly problems within small datasets where complex models are not that efficient. However, the efficiency of KNN can be a concern in large IoT networks since it involves storing and comparing all training data points against every test point—functions that are computationally expensive [42].

Autoencoders represent neural networks used for unsupervised anomaly detection. An autoencoder compresses data into a lower-dimensional form and then reconstructs it. Reconstruction errors suggest the presence of anomalies. For instance, in IoT networks, autoencoders will monitor traffic deviations, including DDoS attacks, and find threats from unlabelled data with no obvious indicators at first sight [43].

Long Short-Term Memory networks are a type of RNN that is specially built to learn long-term dependencies in sequences. Generally, the LSTMs are best fitted for the detection of temporal anomalies in the IoT networks, like spikes in network traffic or changes in device behavior. This will be quite effective in monitoring message sequences of MQTT communications for any deviation that may signal DDoS attacks [43].

Fuzzy logic is a machine-learning technique that performs well under uncertain or imprecise information; therefore, it will be helpful in IoT security. It will be able to adapt security mechanisms w.r.t. changing threat level, network load, or connected devices. Fuzzy logic becomes especially helpful in diversified IoT ecosystems, where devices and communication protocols are different and defining strict security rules is hard to define [44].

Hybrid approaches utilize the strengths of multiple machine learning techniques for securing IoT. For example, the complexity in the pattern may be well detected by neural networks, but decision trees will give real-time decisions. A combination of supervised and unsupervised learning will detect known attacks and highlight new threats. These hybrid systems perform well in securing IoT networks with a large variety in IoT devices and protocols [45].

These different machines learning methods, including autoencoders, LSTMs, fuzzy logic, and hybrid techniques, form a wide toolset that secures IoT networks, with most of them employing the MQTT protocol. Through the careful exploration and leveraging of each strength, IoT systems will be able to detect DDoS attacks at runtime and mitigate them to ensure safe and secure communication in light of evolving threats.

F. Decision Tree for MQTT Security

I will design a model based on Decision Trees for the identification of DoS attacks in MQTT communication. Decision Trees are a simple, effective way to classify network traffic and recognize anomalies that may indicate potential security threats. Decision Trees represent a widely utilized algorithm within the domain of machine learning, predominantly employed in supervised learning scenarios, wherein the model undergoes training on labeled datasets to facilitate predictions or classifications. Basically, a Decision Tree is partitioning the dataset into small subsets according to one of the most critical features at each stage, hence creating a tree-like configuration. In such a structure, every internal node shows a decision about an attribute; every branch indicates the result of that decision, and every leaf node is a definitive class label or output. It is a process for determining the best attribute to split data based on some criteria, which can include Gini impurity and information gain. Both these criteria measure how good the feature is to separate the data. In the creation of the tree, recursive partitioning of the data continues until data can no longer be divided, or a stopping criterion is reached. Major advantages that come with Decision Trees are simplicity, interpretability, and the handling of both numerical and categorical data. However, they are prone to overfitting in the case of a too-big tree; this is normally mitigated with techniques such as pruning. Hence, the basic steps for implementing a decision tree mainly pertain to dataset preparation, the selection of a splitting criterion, recursive partitioning of data, and the integration of a trained model into a decision-making system. The very intuitiveness and graphical display of Decision Trees make them fairly applicable to a beginner, since they represent a transparent progress from the input data to the outcome decision.

Decision Trees have been applied considerably within the realm of a wide variety of intrusion detection systems. In this system, Decision Trees have been introduced that categorize network traffic into benign or nonbenign activities with the use of attributes like connection duration, protocol type, and service type. Some of the attributes of the system include the handling of both categorical and continuous data, while at the same time providing a transparent, see-through decision-making structure that could easily be understood and checked by cyber-security personnel. Another example is using decision trees in an IDS for detecting DoS attacks in MQTT communication. In this case, it was designed to monitor MQTT messages in order to recognize, by their patterns in message size, frequency, and source, possible anomalies that may indicate a DoS attack. Deviations from normal behavior could be efficiently detected using a Decision Tree model and allowed for appropriate alerting and responding in good timing. The Decision Tree model was quite simple and was, therefore, relevant to applications that required real-time operations.

IDS, currently based on the Decision Trees proves that the algorithm is able to detect and respond to security threats under various environments. Such systems are valued because of high accuracy in the identification of intrusions, ease of implementation, and capabilities for yielding actionable insights into network security. Also note that although Decision Trees have many benefits, they are heavily tuned or optimized by applying methods like pruning for avoiding overfitting and ensuring practical performance in complex network contexts with ease.

III. METHODOLOGY

The project, from the very outset, had planned on making use of Azure IoT Hub in cooperation with other services such as Azure Functions and Stream Analytics in building a secure and scalable IoT infrastructure. It was to make use of the said services for processing data in real time, integrating machine learning, and sending automated responses. Yet, there are a few limitations, which indeed required modification to the above-mentioned original approach due to some obvious limitations-like the AzureIoTHubMQTTClient library being tailor-made only for ESP8266 but incompatible with the ESP32 module, and some of the crucial services not available in the free tier like Azure Functions. This drive into implementing alternative solutions using Firebase and Node-RED to enable one to continue doing real-time data processing and machine learning.

The methodology begins with the incorporation of a microcontroller ESP32, acting as a key data acquisition module in the IoT architecture. An ESP32 is able to send real-time data through the MQTT protocol in Node-RED with great efficiency and is capable of thorough processing and analysis of that data. And for storing data utilize Firebase Realtime database. Besides improved capabilities for data management, this is one of those architectures which can provide full support to the integration of higher machine learning techniques. In addition, a machine learning model is created in Google Colab with much deliberation, by training and deployment of robust datasets on both normal operational traffic and traffic patterns indicative of DDoS attacks. By the end of the training process, the model will be integrated into Node-RED for real-time monitoring of continuous data feeds to rapidly detect a possible DDoS threat and trigger an alert to the dashboard upon detection.

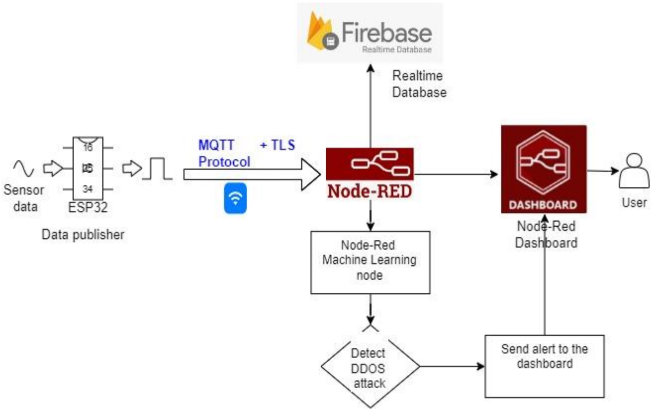


Figure 3. Process Flow

The first process of this project is the setup of the ESP32 microcontroller, to be integrated with a voltage sensor for real-time electrical parameter monitoring. Hook up the voltage sensor to the ESP32, making sure that wiring is proper to attain valid data acquisition (Figure 4). Program the ESP32 using the Arduino IDE (Figure 5). The code first includes the required libraries for Wi-Fi and MQTT functionalities: respectively, `WiFi.h` and 'WiFiClientSecure', and `PubSubClient.h`. Now, to connect to the Wi-Fi, the code defines the SSID and password of the network it will use to connect to the internet. The code defines the MQTT broker details, the address, and the topics of interest. There are open-source MQTT brokers available like Mosquitto, HiveMQ Community Edition, and EMQX to name a few. Here, for this project, I will use HiveMQ as the broker for MQTT to efficiently connect data between ESP32 and Node-RED. ESP32 reads the voltage continuously from that sensor once configured, converts into a suitable message structure in a Json payload message, and publishes the same data to the named MQTT topics. This enables Real time transmission of data in MQTT communication.

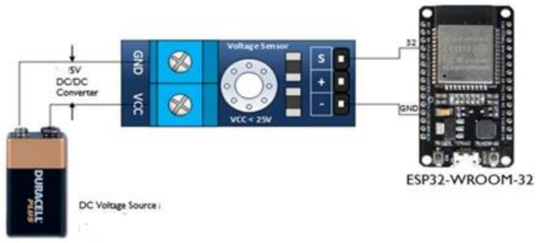
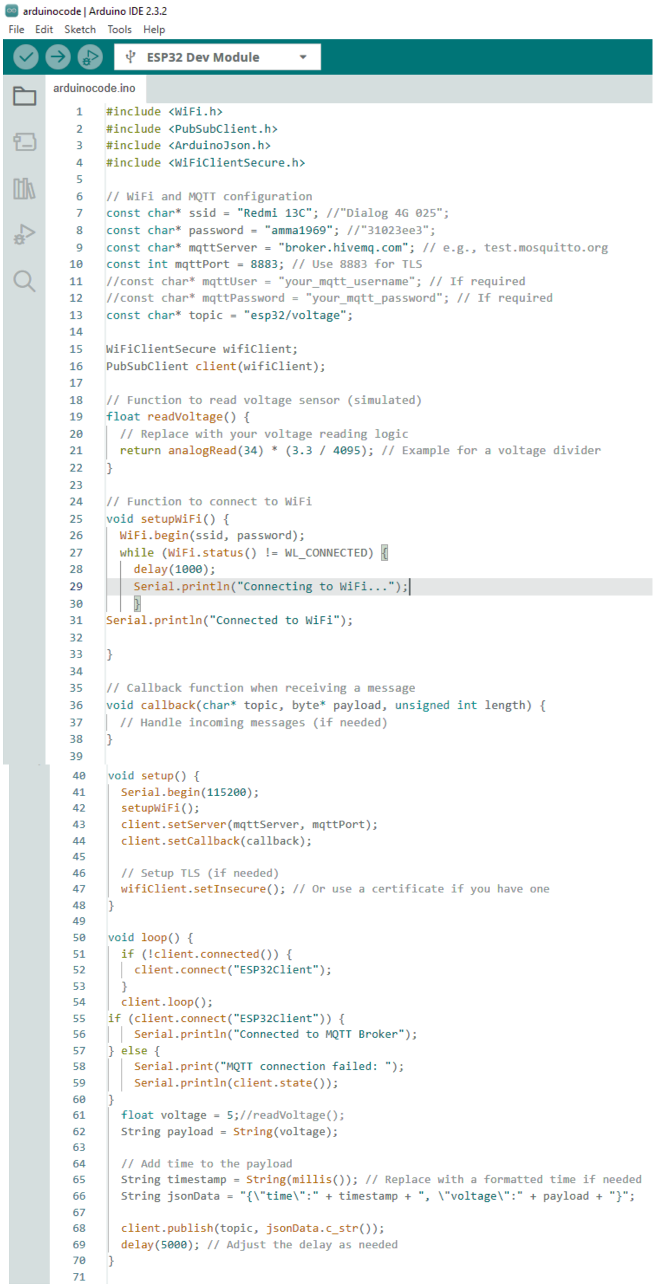


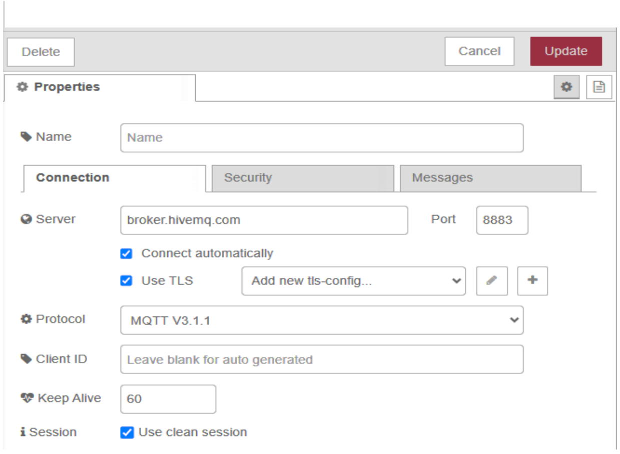
Figure 4. Hardware Setup

The next part of the project consisted of the configuration of Node-RED, which receives and processes data sent by the ESP32 using the MQTT protocol. Node-RED is an open-source flow-based development tool installed and set up to serve as a bridge between the MQTT broker and the system that processes the data. Inside the Node-RED interface, an MQTT input node subscribed to the desired topic to which the ESP32 published. HiveMQ broker acts like an intermediary that relays information between ESP32 and the MQTT subscriber. TLS enabled to encrypt the data sent from ESP32 to the MQTT broker and vice-versa for at least partial data confidentiality and integrity (Figure 6). Upon establishment of the MQTT connection (Figure 7), data from ESP32 was captured in real-time from Node-Red.



```
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <WiFiClientSecure.h>

// WiFi and MQTT configuration
const char* ssid = "Redmi 13C"; //"Dialog 4G 025";
const char* password = "amma1969"; //"31023ee3";
const char* mqttServer = "broker.hivemq.com"; // e.g., test.mosquitto.org
const int mqttPort = 8883; // Use 8883 for TLS
//const char* mqttUser = "your_mqtt_username"; // If required
//const char* mqttPassword = "your_mqtt_password"; // If required
const char* topic = "esp32/voltage";

WiFiClientSecure wifiClient;
PubSubClient client(wifiClient);

// Function to read voltage sensor (simulated)
float readVoltage() {
  // Replace with your voltage reading logic
  return analogRead(34) * (3.3 / 4095); // Example for a voltage divider
}

// Function to connect to WiFi
void setupWiFi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
    }
Serial.println("Connected to WiFi");

}

// Callback function when receiving a message
void callback(char* topic, byte* payload, unsigned int length) {
  // Handle incoming messages (if needed)
}

void setup() {
  Serial.begin(115200);
  setupWiFi();
  client.setServer(mqttServer, mqttPort);
  client.setCallback(callback);

  // Setup TLS (if needed)
  wifiClient.setInsecure(); // Or use a certificate if you have one
}

void loop() {
  if (!client.connected()) {
    client.connect("ESP32Client");
  }
  client.loop();
if (client.connect("ESP32Client")) {
    Serial.println("Connected to MQTT Broker");
} else {
    Serial.print("MQTT connection failed: ");
    Serial.println(client.state());
}
  float voltage = 5;//readVoltage();
  String payload = String(voltage);

  // Add time to the payload
  String timestamp = String(millis()); // Replace with a formatted time if needed
  String jsonData = "{\"time\":" + timestamp + ", \"voltage\":" + payload + "}";

  client.publish(topic, jsonData.c_str());
  delay(5000); // Adjust the delay as needed
}

```

Figure 5. Arduino IDE code



Name: Name

Server: broker.hivemq.com Port: 8883

☑ Connect automatically

☑ Use TLS: Add new tls-config...

Protocol: MQTT V3.1.1

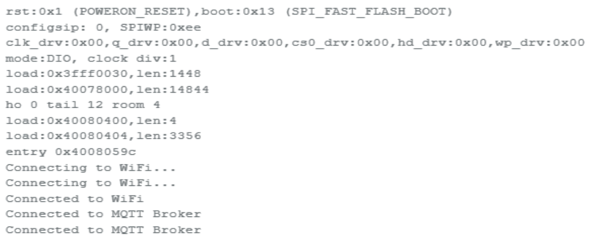
Client ID: Leave blank for auto generated

Keep Alive: 60

Session: ☑ Use clean session

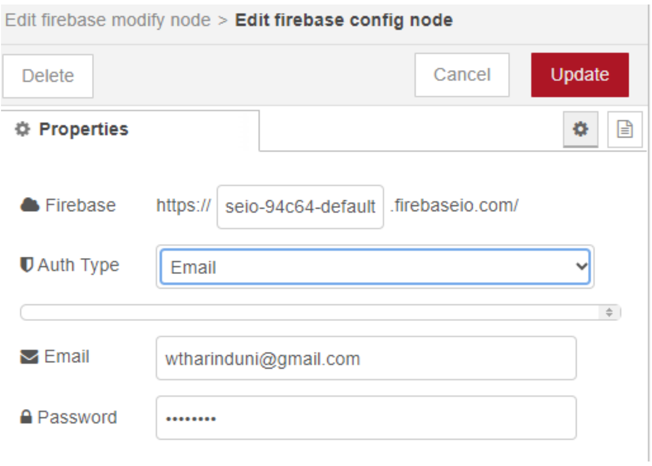
Figure 6. MQTT configuration

The next procedure was the setup of Node-RED to send data to Firebase Realtime Database storage. This entailed the creation of a Firebase project and, subsequently, the generation of credentials comprising the URL for the database and authentication. This connection in Node-RED used the nodes for Firebase to allow structuring of incoming data so that it could be suitable for storage. Security was enhanced through email and password authentication for secure access to the Firebase Realtime Database. This would imply that users would log in using credentials to gain access to the database, thereby securing the data against unauthorized access. The Firebase Realtime Database, once set up, receives data from the processed ESP32 effectively and ensures good storage of such data for easy retrieval upon analysis in the near future.



```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1448
load:0x40078000,len:14844
ho 0 tail 12 room 4
load:0x40080400,len:4
load:0x40080404,len:3356
entry 0x4008059c
Connecting to WiFi...
Connecting to WiFi...
Connected to WiFi
Connected to MQTT Broker
Connected to MQTT Broker
```

Figure 7. MQTT connection establishment



Firebase: https:// seio-94c64-default .firebaseio.com/

Auth Type: Email

Email: wtharinduni@gmail.com

Password: ••••••••

Figure 8. Firebase configuration

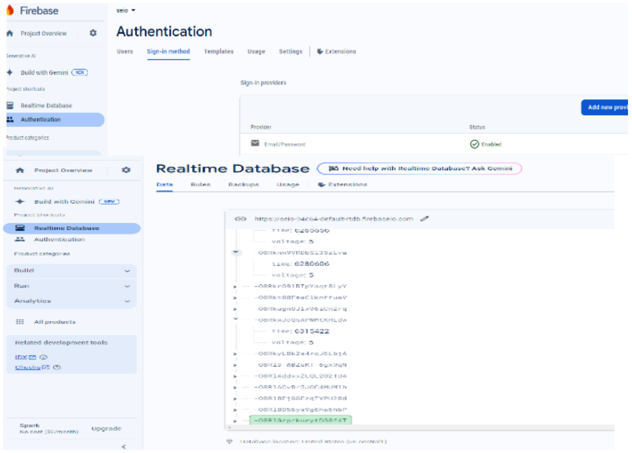
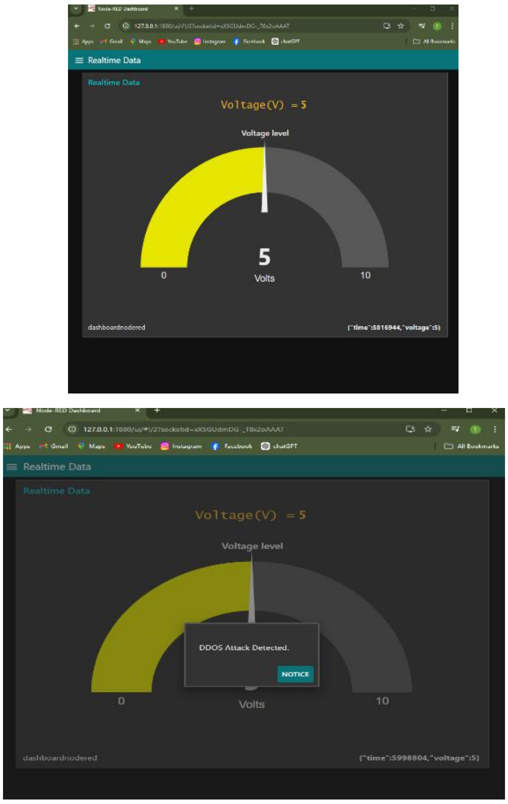


Figure 9. Realtime Database setup

The next step is to prepare Node-RED for creating a dashboard, just to visualize and monitor in real time some data from ESP32. The dashboard is an interactive interface; it shows in real time the voltage reading. Apply ready visualization nodes such as text, gauges, and notification to let the user immediately assess the status of the parameters in monitoring. It also displays, on the dashboard, critical alerts and notifications of DDoS attack detections for further timely responses. This will represent centralized data visualization that enhances user experience and supports effective decision-making through clear, concise visualization of operational metrics for the system.



{"time":5816944,"voltage":5}

DDOS Attack Detected.

{"time":5998804,"voltage":5}

Figure 10. Dashboard

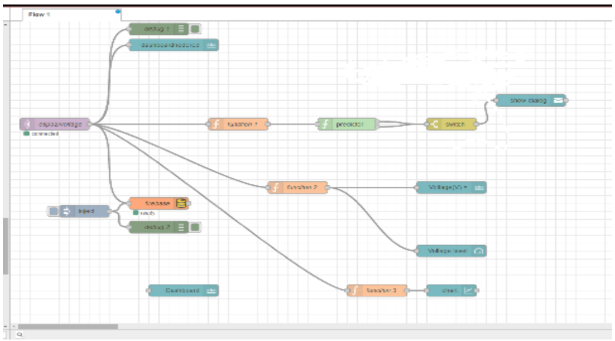


Figure 11. Node-Red Node settings

Subsequently, the methodology of this project involved the training and integration of a machine learning model in detecting DDoS attacks within the incoming MQTT messages. The model was to be designed and trained by using Google Colab with a dataset sourced from Kaggle. The dataset consisted of labeled instances of normal traffic and DDoS attack patterns. Labeling of data was an important part since it has allowed the model to learn the features distinguishing normal and attack traffic. Various algorithms were tested, and the most appropriate model for the detection of anomalies was chosen. Once the model had been trained, it was exported with a view to deployment in Node-RED. A flow was then created, dedicated to integrating the model such that incoming MQTT messages from the ESP32 would be analyzed in real time. It was designed to provide alerts if the traffic patterns went way out of established norms. This would, in turn, offer proactive monitoring to allow for immediate responses whenever there could be a potential DDoS attack for improved security of the system.

Some major resources were employed in the project for the realization of the aim of the research. The hardware components involved the ESP32 microcontroller, interfaced with a voltage sensor that is necessary for data acquisition. For development, Node-RED was the main tool used for flow-based programming, while Firebase was utilized for the secure data storage. Programming languages included Arduino IDE for programming the ESP32 and JavaScript for configuration in Node-RED. The project also applied communication via the MQTT protocol with enabled TLS, which makes data transmission secure. In the end, the training of the machine learning model was performed using Google Colab on a properly pre-processed dataset taken from Kaggle for the purpose of effective detection of DDoS attacks.

IV. Results and Discussion

The proposed system demonstrates promising performance regarding the detection of DDoS attacks in IoT environments over communications using MQTT. In this case, the ESP32 microcontroller will collect the voltage data coming from sensors and send it to Node-RED via MQTT transmission. The integration of TLS provides secure data transmission. Accordingly, Node-RED parses and allows visualization in real time, neat interfaces for voltage and system performance monitoring of incoming data. It efficiently identified the anomalies. The machine learning model, trained with labeled data from Kaggle, showed promise but was a bit difficult during training due to certain misalignments between features in normal packet data and DDoS packet data. So, because there was this discrepancy during the learning process, there would be a bit of latency in detection. It generated an alert when it had detected an anomaly. It was able to catch the DDoS threat in due time, taking remedial action against the system.

In all, the integration of machine learning into IoT frameworks is very promising for developing enhanced security measures; however, challenges noted create demands for further refinement in feature selection and continued training in order to improve the detection accuracy and responsiveness in future versions.

V. Conclusion

This project eventually realized a very excellent DDoS attack detection system in MQTT communication based on machine learning and IoT technology. The results showed that good detection performance was achieved while major challenges needed to be concerned about the data alignment, training and latency. Future work should devote energy to refining feature selection, enhancing the model training process, and digging out more advanced techniques like ensemble learning for much higher accuracy and responsiveness in attack detection, to enhance the safety of IoT networks against ever-evolving threats.

VI. References

[1] N. Moustafa, B. Turnbull and K. -K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," *IEEE Internet of Things Journa,* vol. 6, pp. 4815-4830, 2019.

[2] I. Vaccari, S. Narteni, M. Aiello, M. Mongelli and E. Cambiaso, "Exploiting Internet of Things Protocols for Malicious Data Exfiltration Activities," *IEEE Access,* vol. 9, pp. 104261-104280, 2021.

[3] A. R. Heinrihs Kristians Skrodelis, "Synthetic Network Traffic Generation in IoT Supply," *International Scientific Conference on Information Technology and Management Science of Riga Technical University,* 2022.

[4] D. J. M. R. A. M. S. A. M. MOHAMMADALRAZIB, "Cyber Threats Detection in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework," vol. 10, 2022.

[5] N. H. P. D. R. Z. Lourdes Cecilia Ruiz Salvador, "SCADA Systems: Security Concerns and Countermeasures," *IEEE 21st World Symposium on Applied Machine Intelligence and Informatics,* 2023.

[6] I. Šenk, S. Tegeltija and L. Tarjan, "Machine Learning in Modern SCADA Systems:," *23rd International Symposium INFOTEH-JAHORINA,* 2024.

[7] A. I. Pricop, M. GavrilaŞ, A. SĂlceanu and B. C.Neagu, "Power systems resilience against cyber-attacks. A systematic analysis," *2023 10th International Conference on Modern Power Systems (MPS),* pp. 1-7, 2023.

[8] S. P. F. P. Riccardo Colelli, "Securing connection between IT and OT: the Fog Intrusion Detection System prospective," *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT),* pp. 444-448, 2019.

[9] Daher and L. Abu, "Towards Secure IoMT: Attack Detection Using," *2023 16th International Conference on Developments in eSystems Engineering (DeSE),* 2023.

[10] Daher and L. Abu, "Towards Secure IoMT: Attack Detection Using Deep Q-Learning in Healthcare Networks," *2023 16th International Conference on Developments in eSystems Engineering (DeSE),* 2023.

[11] M. A. et, "NHS WannaCry Ransomware Attack: Technical Explanation of The Vulnerability, Exploitation, and Countermeasures," *2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI),* pp. 1-6, 2022.

[12] D. -Y. KAO, S. -C. HSIAO and R. TSO, "Analyzing WannaCry Ransomware Considering the Weapons and Exploits," *2019 21st International Conference on Advanced Communication Technology (ICACT),* pp. 1098-1107, 2019.

[13] E. Kirdan, P. Horvath and M.-O. Pahl, "Slow Denial of Service Attack on MQTT-Based IoT," *023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom),* 2023.

[14] M. Ramaiah and M. Y. Rahamathulla, "Securing the Industrial IoT: A Novel Network Intrusion Detection Models," *2024 3rd IEEE International Conference on Artificial Intelligence for Internet of Things (AIIoT 2024),* 2024.

[15] A. Reed, L. S. Dooley and S. K. Mostefaoui, "Packet Filtering and Sampling for Efficient Slow Denial of Service Detection in Resource Scarce IoT Networks," *2023 International Symposium on Networks, Computers and Communications (ISNCC),* 2023.

[16] Kaur, Kulwinder, Ayoade and John, "Analysis of DDoS Attacks on IoT Architecture," 2023.

[17] D. C. Jerrin Simla A, "Review on Application Layer Protocol for IoT enabled Agricultural Intrusion Detection," *Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021),* pp. 1170-1174, 2021.

[18] H. K. Skrodelis and A. Romanovs, "Synthetic Network Traffic Generation in IoT Supply Chain Environment," *2022 63rd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS),* 2022.

[19] U. Garg, S. Kumar and M. Kumar, "A Hybrid Approach for the Detection and Classification of MQTT-based IoT-Malware," *Proceedings of the International Conference on Sustainable Computing and Data Communication Systems (ICSCDS-2023),* pp. 1154-1159, 2023.

[20] P. C. D. T. HARIPRASAD SIDDHARTHAN, "SENMQTT-SET: An Intelligent Intrusion Detection in IoT-MQTT Networks Using Ensemble Multi Cascade Features," vol. 10, pp. 33095-33110, 2022.

[21] Leal, Roberto, L. Santos, L. Vieira, R. Gonçalves and C. Rabadão, "MQTT flow signatures for the Internet of Things," *2019 14th Iberian Conference on Information Systems and Technologies (CISTI),* 2019.

[22] F. Jaafar, Y. Malik, J. Serre, H. Wang and T. Wang, "Lightweight Intrusion Detection in MQTT Based Sensor Network," *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME),* 2022.

[23] A. S. D and P. PRABU, "End-to-End Encryption in Resource-Constrained IoT Device," vol. 11, pp. 70040-70051, 2023.

[24] B. Oniga, S. H. Farr, A. Munteanu and V. Dadarlat, "IoT Infrastructure Secured by TLS Level Authentication and PKI Identity System," *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4),* pp. 78-83, 2018.

[25] A. GARBA, D. KHOURY, P. BALIAN, S. HADDAD, J. SAYAH, Z. CHEN, Z. GUAN, H. HAMDAN, J. CHARAFEDDINE and K. AL-MUTIB, "A Flexible and Efficient Authentication and

Secure Certificates Authentication for IoT Applications," vol. 11, pp. 28370-28383, 2023.

[26] H.Dagale and N. Maheshwari, "Secure communication and firewall architecture for IoT applications," *2018 10th International Conference on Communication Systems & Networks (COMSNETS),* pp. 328-335, 2018.

[27] N. Nasrullayev, S. Muminova, D. K. Istamovich and M. Boltaeva, "Providing IoT Security in Industry 4.0 using Web Application Firewall," *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC),* pp. 1788-1792, 2023.

[28] IMRAN, M. F. ZUHAIRI, S. M. ALI, Z. SHAHID and M. M. S. MUHAMMAD MANSOOR ALAM, "Realtime Feature Engineering for Anomaly Detection in IoT Based MQTT Networks," *25700-25718,* vol. 12, 2024.

[29] C. Huang, Y. Zong, J. Chen, W. Liu, J. Lloret and M. Mukherjee, ""A Deep Segmentation Network of Stent Structs Based on IoT for Interventional Cardiovascular Diagnosis," *IEEE Wireless Communications,* vol. 28, pp. 36-43, 2021.

[30] S. KUMAR, S. GUPTA and S. ARORA, "Research Trends in Network-Based Intrusion Detection Systems: A Review," vol. 9, pp. 157761-157779, 2021.

[31] X. LI, Q. WANG, X. LAN, X. CHEN, D. CHEN and N. ZHANG, "Enhancing Cloud-Based IoT Security Through Trustworthy Cloud Service," vol. 7, pp. 9368-9383, 2019.

[32] A. Kataria, "An ML-Based Intrusion Detection System Design and Evaluation for Enhanced Cybersecurity," *2023 International Conference on Communication, Security and Artificial Intelligence (ICCSAI),* pp. 1036-1039, 2023.

[33] M. Z. Alom and T. M. Taha, "Network intrusion detection for cyber security using unsupervised deep learning approaches," *2017 IEEE National Aerospace and Electronics Conference (NAECON),* pp. 63-69, 2017.

[34] J. L. a. H. P. M. Kwon, "Intelligent IoT Connectivity: Deep Reinforcement Learning Approach," *IEEE Sensors Journal,* vol. 20, pp. 2782-2791, 2020.

[35] JAYALAXMI, P. L. S., R. SAHA, G. KUMAR, M. CONTI and T.-H. KIM, "Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs," vol. 10, pp. 121173-121192, 2022.

[36] FATANI, ABDULAZIZ, ELAZIZ, M. ABD, DAHOU, ABDELGHANI, M. A. A. AL-QANESS and S. LU, "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," vol. 9, pp. 123448-123464, 2021.

[37] I. ULLAH and Q. H. MAHMOUD, "Design and Development of RNN Anomaly Detection Model for IoT Networks," vol. 10, pp. 62722-62750, 2022.

[38] S. Tauqeer, H., Ahmad, S. M. A. I. W.Shah and M.Zaman, "Implementation of Intrusion Detection System in the Internet of Things: A Survey," *2020 IEEE 23rd International Multitopic Conference (INMIC),* pp. 1-6, 2020.

[39] N. Y. S. Y. Mohd Anuaruddin Bin Ahmadon, "Process-Based Intrusion Detection Method for IoT System with MQTT Protocol," *2019 IEEE 8th Global Conference on Consumer Electronics(GCCE),* pp. 953-956, 2019.

[40] Z. AZAM, M. M. ISLAM and M. N. HUDA, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree," *ZAHEDI AZAM , MD. MOTAHARUL ISLAM , AND MOHAMMAD NURUL HUDA,* vol. 11, 2023.

[41] Q. Z. a. Y. Hara-Azumi, "Hardware/Software Codesign of Real-Time Intrusion Detection System for Internet of Things Devices," *IEEE Internet of Things Journal,* vol. 11, pp. 22351-22363, 2024.

[42] A. Qureshi, M. A. Qureshi, H. A. Haider and R. Khawaja, "A review on machine learning techniques for secure IoT networks," *2020 IEEE 23rd International Multitopic Conference (INMIC),* 2020.

[43] H. LIAO, M. Z. MURA, M. K. H. A. H. M. AMAN, J. FANG1, X. HU and A. U. R. KHAN, "A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things," vol. 12, pp. 4745-4761, 2023.

[44] M. B. G. a. R. nski, "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, pp. 24843-24855, 2022.

[45] M. Z. H. M. Z. H. H. J. A. A. J. A. USAMA SHAHID, "Hybrid Intrusion Detection System for RPL IoT Networks Using Machine Learning and Deep Learning," vol. 12, pp. 113099-113112, 2024.

[46] A. DUNMORE, J. JANG-JACCARD, F. SABRINA and J. KWAK, "A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection," vol. 11, pp. 76071-76094, 2023.

[47] D. Rani, N. S. Gill, P. Gulia, F. Arena and G. Pau, ""Design of an Intrusion Detection Model for IoT-Enabled Smart Home," *IEEE Access,* Vols. 52509-52526, p. 11, 2023.

[48] S.Gupta and A. S. Dandotiya, "SSFID: A Survey and Analysis of Security Framework for IoT Devices," *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG),* pp. 1-6, 2023.

[49] R.Kumar and Raju, "Improving IoT Security Through Machine Learning," *2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS),* pp. 926-932, 2023.

[50] N. T. Cam and N. G.Trung, "An Intelligent Approach to Improving the Performance of Threat Detection in IoT," *IEEE Access,* vol. 11, pp. 44319-44334, 2023.

[51] M. M. Mohammed and K. M. A. Alheet, "Deep Learning Model For IDS In the Internet of Things,"

7th International Conference on Contemporary Information Technology and Mathematics (ICCITM-2021), pp. 153-159, 2021.

[52] S. H. S. Ariffin, N. H. Mustaffa, F. Dewanta, I. W. Hamzah, M. A. Baharudin and N. H. A. Wahab, "Hybrid Feature Selection Based Lightweight Network Intrusion Detection System for MQTT Protocol," *2023 15th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pp. 226-230, 2023.