A. P. and K. *EURASIP Journal on Wireless Communications and Networking* (2019) 2019:90
https://doi.org/10.1186/s13638-019-1402-8

EURASIP Journal on Wireless Communications and Networking

REVIEW Open Access

Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things

Haripriya A. P.* and Kulothungan K.

Abstract

The advancements in the domain of Internet of Things (IoT) accelerated the development of new communication technologies. Machine to machine communication in IoT is accomplished with application protocols such as the Constrained Application Protocol and Message Queuing Telemetry Transport (MQTT). The openness of these protocols leads to various types of attacks including DoS Attack. MQTT is widely used in secure IoT applications like health monitoring. One of the prominent attacks in IoT is the denial of service (DoS) attack. This enforces the need for an efficient intrusion detection system method in MQTT-based application. In this paper, we address the vulnerabilities in MQTT, through which intruders can control the low-configuration devices in the network. This paper proposes a lightweight fuzzy logic-based intrusion detection scheme called Secure-MQTT, for detecting malicious activity during the communication between IoT devices. The proposed method uses a fuzzy logic-based system to detect the malicious behavior of the node with the help of a fuzzy rule interpolation mechanism. Secure-MQTT avoids the use of a dense rule base by exploiting the fuzzy rule interpolation that generates rules dynamically. The proposed method provides an effective mechanism to protect the low configuration devices from the DoS attack. The simulation results show that the proposed method detects the attacks more accurately when compared to the existing methods.

Keywords: Intrusion detection, Fuzzy interpolation, Fuzzy inference engine, Rule base, Publisher, Subscriber, Broker

1 Introduction

IoT is a promising future network paradigm that enables communication among heterogeneous smart devices. The number of connected devices is expected to be 50 billion by 2020 [1]. The heterogeneous and huge number of devices in the IoT leads to a difficulty in monitoring the data exchange between the devices, which in turn makes the intrusion detection system (IDS) in IoT a potential research problem. Several protocols such as Message Queuing Telemetry Transport (MQTT) [2], Constrained Application Protocol (CoAP) [3], Extensible Messaging and Presence Protocol (XMPP) [4], and Advanced Message Queuing Protocol (AMQP) [5] are introduced to transfer the message in the IoT network. MQTT is the best candidate for M2M communication due to its lightweight characteristics and ability to work efficiently in low-power, limited memory devices as compared to its counterpart, CoAP [6]. This paper focuses on IDS in the MQTT protocol.

The MQTT brokers are considered as the main component of any MQTT-based IoT application as it offers many services to the clients [7]. The main vulnerability of MQTT protocol is flooding the broker which leads to a DoS attack [8]. The attacker compromises the broker and sends false control or data packets during the DoS attack. Therefore, automatic recoverability from the DoS attack, the time taken for the recoverability, and the impact of broker failure in the IoT application are the significant security concerns in MQTT protocol.

One of the countermeasures for a DoS attack in MQTT is a certificate based on SSL/TLS authentication [9], which is not advisable for IoT devices since certificate management increases the computation and communication overhead. Also, session key generation and distribution in SSL/TLS reduces the performance of MQTT. Another security measure to reduce the amplification of the DoS

* Correspondence: haripriya@auist.net
Department of IST, Anna University, Chennai, Tamil Nadu, India

attack is throttling [10] which prevents the attacker from the identification of the frequently subscribed topic and flood with false messages in the broker. The throttling is inadequate in providing security for IoT environments because of its inefficiency to withstand against a large-scale DoS attack. The chance of discarding important messages may occur during throttling which should be avoided. In a botnet attack, an intruder controls the devices in IoT by compromising the broker by installing malware on a compromised node [11]. The methods proposed to detect and prevent the attacks discussed above adopt the traditional IDS which will not produce effective results in all IoT network conditions. This is mainly because of the dynamic network features of IoT and low configuration IoT devices. This shows the demand for a lightweight IDS for MQTT to secure the communication among the constrained devices in IoT.

In this paper, a novel lightweight IDS is proposed for MQTT-based IoT applications using fuzzy logic. The proposed system identifies the network anomalies with the help of fuzzy variables. The degree of anomalous behavior of the node is determined from this fuzziness. The fuzzy inference system executes IF-THEN-based fuzzy rules that are used to define the different network conditions to detect the attacks. The proposed fuzzy-based IDS in MQTT is the first IDS in its nature in the literature which prevents a DoS attack effectively in IoT applications. Our simulation analysis shows that Secure-MQTT is suitable for smart environments.

The major contributions of this paper are the following:

1. The proposed Secure-MQTT identifies the malicious behavior of the publish-subscribe nodes in the MQTT protocol in the IoT environment. We designed a novel efficient IDS using fuzzy logic which is applied to selected network traffic features. The proposed system employs efficient data preprocessing and a simple rule base for detecting a DoS attack.
2. Subsequently, a lightweight DoS attack detection scheme by employing fuzzy rule interpolation on the rule base is proposed. The fuzzy rule interpolation dynamically updates the rule base based on the past scenario. This enables the increase of the performance of the intrusion detection method.
3. Finally, the proposed IDS is validated and verified under varying network scenarios.

The rest of the paper is organized as follows. Section 2 gives a brief idea about the methods used in Secure-MQTT. Section 3 describes the existing works in IDS detection in IoT networks. Section 4 gives problem formulation and Section 5 illustrates the proposed IDS architecture. The performance analysis of Secure-MQTT is described in Section 6. The conclusions and future work are given in Section 7.

2 Methods

The proposed Secure-MQTT detects the malicious activity of nodes in MQTT broker using the proposed fuzzy logic-based intrusion detection approach. The network traffic behavior of MQTT publishers is monitored, and selected traffic features are trained over a period. The attacking scenario is simulated in the network where the number of the malicious nodes is 10–50% of the total number of nodes deployed in the network. The fuzzy logic-based approach is applied to the selective traffic features. The malicious node is detected according to the fuzzy rules in the rule base. Then, the fuzzy inference engine decides whether the MQTT message has to be accepted or not. The complexity of the fuzzy model is reduced by dynamic fuzzy interpolation methods. It also helps to improve the efficiency of the revised rule base which in turn increases the overall performance of the IDS. The proposed work becomes lightweight in the absence of a dense rule base with dynamic interpolated rules that are derived from the most relevant network traffic features in MQTT-based communication. The Secure-MQTT is compared with existing MQTT-S, where the security is provided with SSL/TLS. The performance analysis shows that the Secure-MQTT outperforms MQTT-S.

3 Related works

This section discusses the relevant proposed works in IDS for IoT networks. In general, unauthorized access to the IoT network should be prevented in time by considering the constraint characteristics of the IoT devices. Traditional network IDS are not compatible with IoT due to the inadequate storage and computing of the devices. IDS can be broadly classified into two, namely anomaly-based IDS and signature-based IDS [12].

3.1 Signature-based IDS

In signature-based IDS, the behavior of the system is compared with the previously known attack patterns. The authors presented a signature-based lightweight IDS, Snort, for low power networks [13]. The pattern matching algorithm used in Snort is the Boyer-More algorithm, which has high efficiency if there is a set of unique pattern match set. However,

the attack pattern in the database should be updated over a period but is not illustrated in the paper. In [14], an attack pattern matching engine, with auxiliary shifting and early decision, is incorporated for better performance of constrained devices in IoT. The auxiliary shifting method avoids the matching operations to increase the computation efficiency.

3.2 Anomaly-based IDS

Anomaly-based IDS monitors the network behavior and classifies the network activity as normal or anomalous. A botnet attack-detecting mechanism for 6LoWPAN gateways is proposed in [15], in which the detection module analyzes the behavior of the network traffic. The botnet attack can be easily launched in the network via sending spam email, information theft, and a DOS attack. An attacker controls the network by compromising the nodes in the network in a botnet attack. The authors have evaluated the detection rate but did not discuss the performance overhead of the detection scheme.

The authors in [16] suggested a sinkhole detection scheme for secure routing on RPL for IoT networks. In this work, the detection scheme observes received and transmitted packets periodically and calculates intrusion ratio. Due to the high false-positive and false-negative rates and more energy consumption, the solution for a sinkhole attack proposed fails for IoT application.

A game theory-based hybrid IDS for low constrained devices is proposed in [17]. In this paper, authors have modeled a game for the normal user and attacker and Nash equilibrium value which determines the anomaly behavior which is also calculated. In [18] is the proposed security information and event management-based IDS for M2M communication. In this work, the detection scheme observes the security events in the networks and applies the correlation method to identify the attack. The system can be improved by adding more libraries for improving detection accuracy.

A hybrid intrusion detection scheme, based on signature and anomaly detection, is proposed in [19]. The scheme has the benefit of storage efficiency of signature-based detection and computing efficiency of the anomaly-based detection. The placement strategy adopted here is centralized and distributed in which IDS scheme is implemented in both constrained devices and 6LOWPAN border router. The work can be extended to find more attacks in the IoT network.

The chance of false detection rate is more in signature-based schemes if the training set does not contain the observed behavior which is having normal characteristics [20]. The signature-based schemes fail to address the continuous data streaming from many heterogeneous IoT devices whereas the abnormal behavior of the network can be effectively detected using machine learning techniques. Therefore, this proposed work adapts anomaly-based detection to find malicious behavior in MQTT-based IoT networks. IDS in MQTT communication is not addressed effectively in literature; although, it is essential for secured IoT applications like military and smart environments. A variety of IDS schemes in IoT that are discussed above use different machine learning models such as artificial neural networks, genetic algorithm, and fuzzy logic which are applied for detecting the anomaly in the IoT network. In order to classify the uncertain and nonlinear data in MQTT, the proposed scheme adopts the fuzzy logic model, which helps to produce better accuracy.

The existing security mechanism in MQTT is implemented through SSL/TLS. The certificate generation and session key management in SSL/TLS increase the computation complexity of the constrained IoT devices. The proposed Secure-MQTT uses a smaller comprehensive set of rules since the fuzzy interpolation method generates rules dynamically. Therefore, the proposed method is light in terms of the computation steps involved.

4 Problem formulation

4.1 Background

MQTT is a widely used application layer protocol to transmit data among the devices in IoT, because of its simplicity and scalability [21]. Figure 1 shows the general MQTT message model. In this model, publisher, subscriber, and the broker are the basic elements for accomplishing communication between the IoT devices [22].

MQTT follows a TCP-based connection establishment procedure. The device sends MQTT a request message, *CONNECT*, to connect with the broker. Once the request is received, the broker will send the acknowledgment, *CONNACK*, to the sending device. Later, the IoT device sends or publishes the message on a particular topic to the broker, and the receiving devices subscribe the messages from the broker. When a request arrives at the broker, the proposed Secure-MQTT analyzes the traffic and compares the stored fuzzy rules to check the presence of an intrusion.

4.1.1 Threat model

Threat model in MQTT aims to identify the attacks against the MQTT broker by analyzing publish-subscribe messages. The proposed system assumes that the malicious devices gain access to the network and these malicious devices prevent the services offered by the broker during publishing and subscribing the messages.

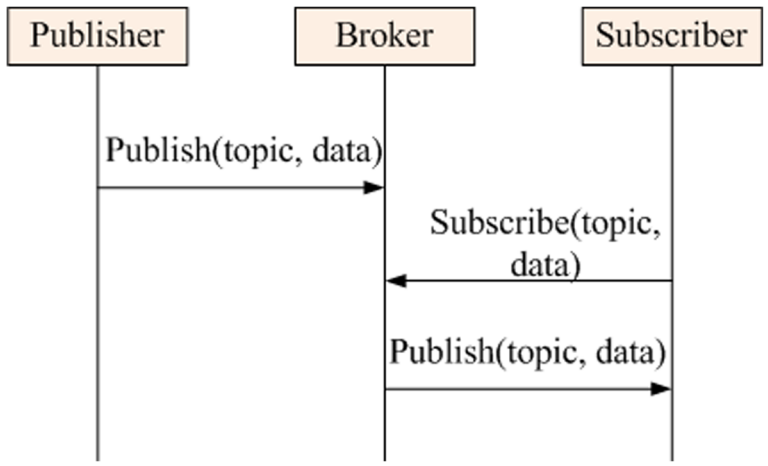

Fig. 1 MQTT Publish/Subscribe topic message model [22]

4.2 Attack scenario

The DoS attack scenario considered in this work is graphically represented in Fig. 2. The network traffic coming to the MQTT broker has to be analyzed.

An attacker can launch a DoS attack in the broker by sending many connection requests continuously thereby making the broker busy as in a flooding attack. If there are many connection requests that arrive at the same time, then the buffer will be drained and the broker will not be in a position to handle new incoming requests. Moreover, the broker is not able to differentiate the normal CONNECT and the spoofed CONNECT message packets. On receiving the flood request messages, the broker starts to acknowledge with CONNACK message. There is an abrupt rate of increase in the number of CONNECT and CONNACK packets during the DoS attack. This results in blocking the broker service and prevents the functioning of the intended IoT network.

4.3 Design goals

Based on the above discussions, we formulate the primary goals of the proposed as follows:

1. To develop a lightweight IDS to detect and prevent the DoS attack in MQTT for IoT devices
2. Design efficient methods to achieve early and timely detection of the flooding of publisher and subscriber messages which results in a DoS attack

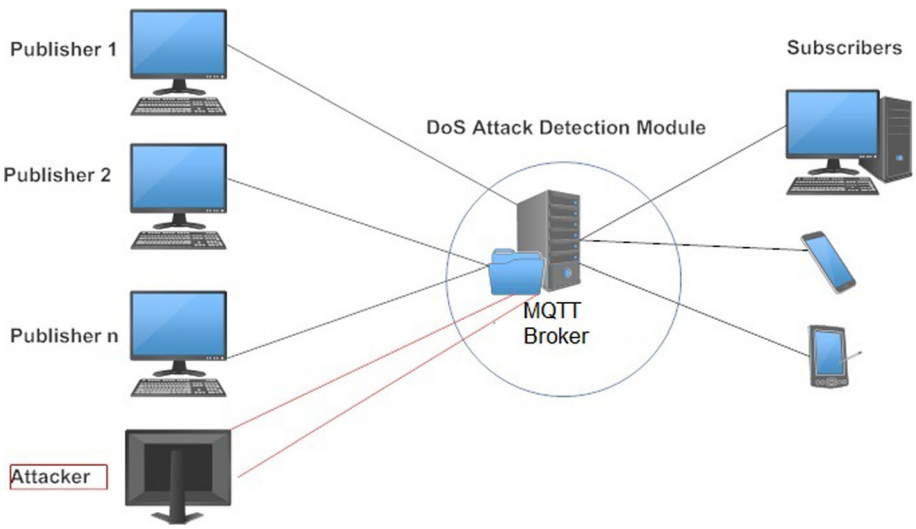

Fig. 2 Attack Scenario in the proposed Secure-MQTT

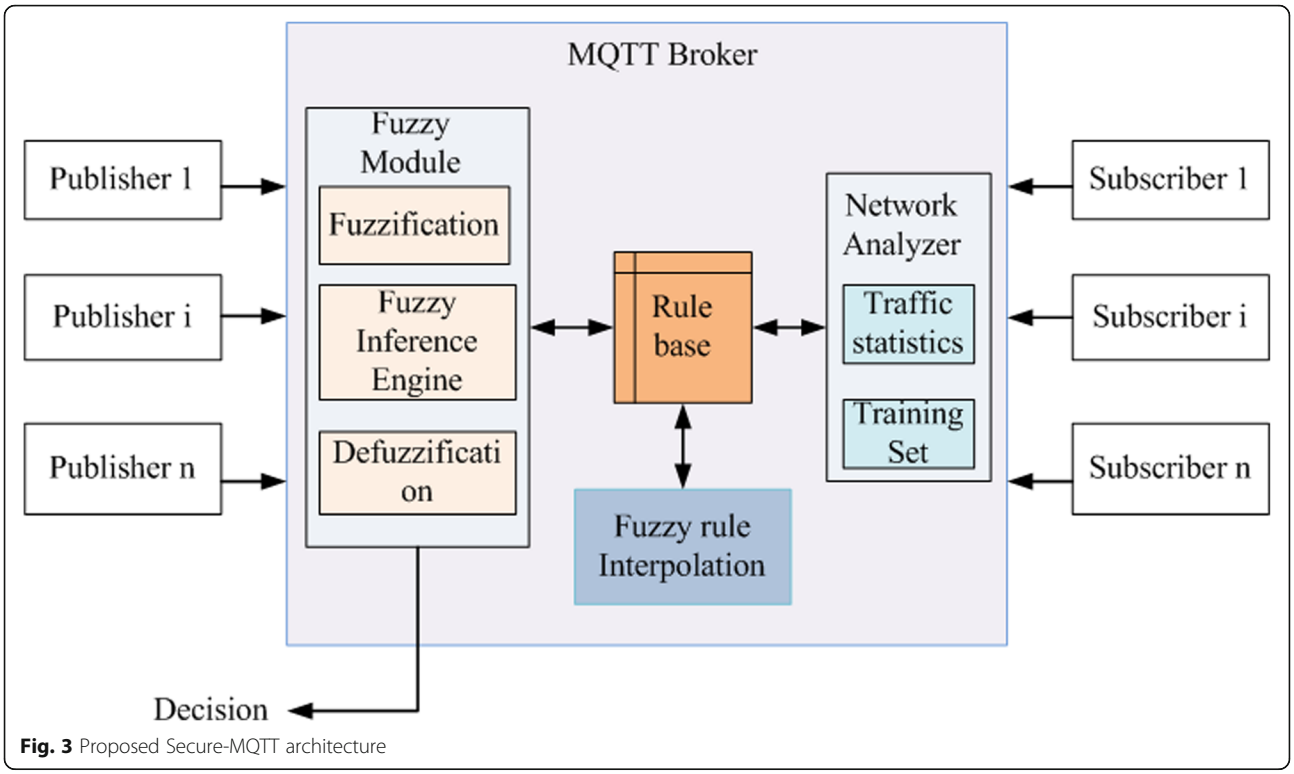

Fig. 3 Proposed Secure-MQTT architecture

3. Design lightweight rule base to enable computationally efficient decision making in fuzzy inference engine

5 Proposed IDS architecture

To achieve the design goals discussed above, a novel fuzzy logic-based IDS is proposed for the detection of an anomaly during the machine to machine communication using the MQTT broker. Figure 3 shows the proposed IDS architecture.

The major components of the proposed Secure-MQTT broker are a fuzzy module, rule base, fuzzy rule interpolation, and network analyzer which are explained as follows:

- The fuzzy module consists of fuzzy computation and fuzzy verification sub-modules namely fuzzification, fuzzy inference engine, and defuzzification. Fuzzification estimates the fuzziness of the MQTT message traffic analysis of the device based on Connection Message Ratio (CMR) and Connection Acknowledgment Message Ratio (CAMR). The fuzzy rule engine selects the appropriate rule and decides whether the request is a legitimate request or not. Defuzzification converts the fuzzified inputs into crisp output.
- Rule base stores the fuzzy rules which are formulated from the training network traffic dataset.
- Fuzzy rule interpolation [23] is applied to the rule base to reduce the complexity in the fuzzy inference engine by deriving new appropriate rules in the rule base.
- The network analyzer consists of traffic statistics and training dataset. The traffic statistics store the history of network traffic behavior for specified time frames. The training dataset stores network traffic features and it is used to train the fuzzy module.

The fuzzy-based DoS detection in the proposed Secure-MQTT is as shown in Fig. 4.

Table 1 Network Traffic Features

Feature name	Description
Connect	Connect command
ConnectAck	Acknowledgment to Connect command
ConnectRate	Percentage of Connect requests arrived at broker
ConnectAckRate	Percentage of ConnectAckRate
PublishMessage	Publish message from publishing client to broker
ConnAck	Request for subscribing message
DisconnectReq	Request to disconnect

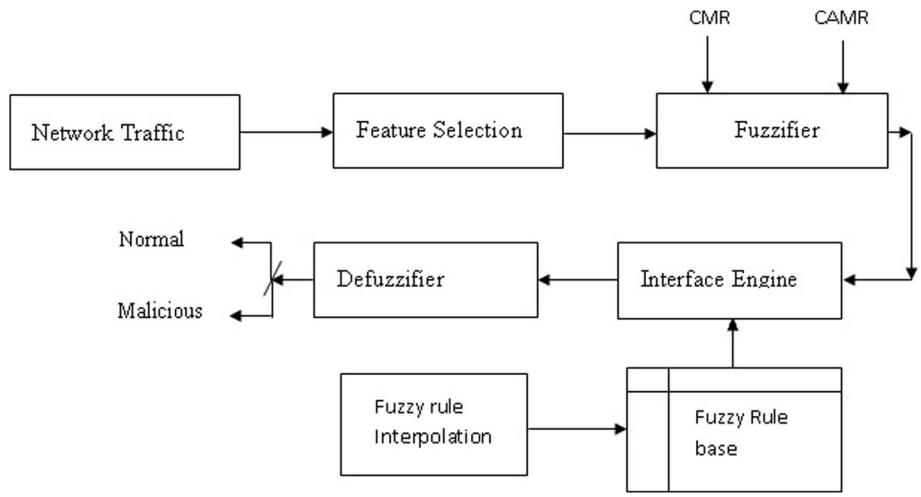

Fig. 4 Fuzzy based DoS detection in proposed Secure-MQTT

5.1 Feature selection

The features used in generating fuzzy rules are selected from the network trace dataset. During this feature selection process, devices collect the required features from the network traffic which is shown in Table 1.

The proposed work considers the features Connect and ConnectAck of CONNECT and CONNACK messages, respectively, for training the proposed system. Then, the next step is to fuzzify the two important variables, CMR and CAMR. The fuzzy variable CMR indicates a fraction of connection requests from the publisher, which is formally given in Eq. 1.

$$\mathrm{CMR} = \frac{N_{\mathrm{connect}}}{N}, \tag{1}$$

where N_{connect} represents the count of the Connect feature and N is the total number of MQTT messages. CAMR represent the fraction of subscriber requests received in the broker which is defined in Eq. 2.

$$\mathrm{CAMR} = \frac{N\mathrm{ConnAck}}{N} \tag{2}$$

where N_{ConnAck} gives the number of connection acknowledgment messages from MQTT broker. The CMR and CAMR are given as inputs to the fuzzy system. The algorithm for IDS using Fuzzy logic is formally stated as follows:

Algorithm 1 IDS using Fuzzy-Logic for MQTT Broker

Input: CMR, CAMR

Output: Check anomaly or not

```
BEGIN
    Fuzzify the input variables using fuzzy membership function
    Derive a fuzzy classifier to generate fuzzy rules which are in the form of IF THEN statement
    Activate the selected rules for a given instance
    Apply fuzzy rule interpolation if the rules are not found in rule base
    Compute the fuzzy output distribution by combining the activated rules in the rule base
    Defuzzify the fuzzy output to obtain a crisp value
END
```

The proposed intrusion detection works on the basis of Algorithm 1. The input variables CMR and CAMR are fuzzified using fuzzy membership function, then derive a fuzzy classifier to generate fuzzy rules which are in the form of an IF-THEN statement. The fuzzy inference engine activates the selected rules in the rule base for a given instance. If no such rule is found in the rule base, then fuzzy rule interpolation that generates rule dynamically is applied. Defuzzify the output variable *anomaly* to obtain a crisp value. The detailed workings of membership function and fuzzy inference engine, fuzzy rule interpolation, and defuzzification are given in the following sections.

5.2 Membership function

The membership values of CMR and CAMR are derived from the observed range of values and computed from the average value of each parameter in the interval [low,

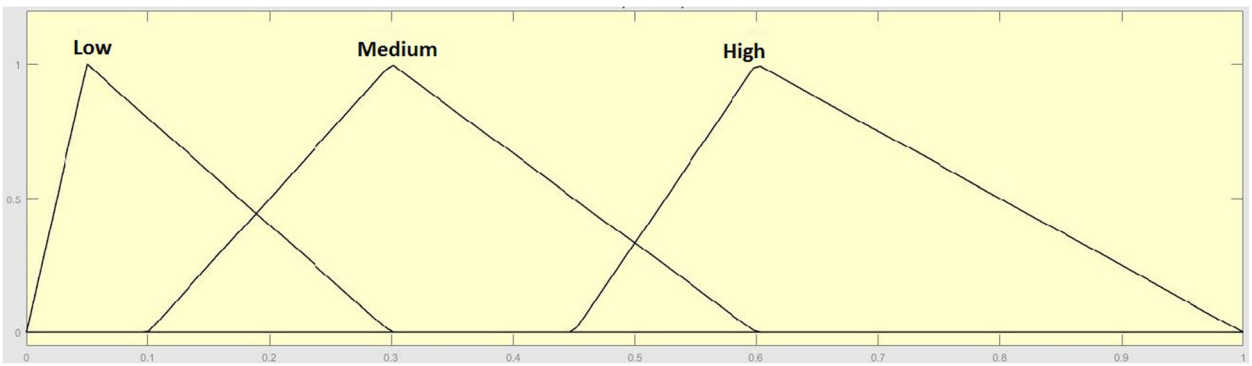

Fig. 5 Membership function of CMR

medium, high] of the training dataset. The output variable anomaly is also a fuzzy variable.

The membership values of the CMR and CAMR are derived from the observed values of each input parameters.

Fig. 5 illustrates the fuzzy set of CMR input variable. The linguistic variables for this fuzzy set are high, medium, and low.

The input variable CAMR is depicted in Fig. 6. The linguistic variables low, medium, and high in CMR and CAMR are represented with a triangular membership function and the medium is represented using triangular membership function. The fuzzy inference engine computes the anomaly computation of each client.

The fuzzy set of output variable *anomaly* is given in Fig. 7. Here, the used linguistic variables are normal, abnormal and attack. The triangular membership function represents the linguistic variables normal, abnormal, and attack.

The MQTT broker in the network monitors the traffic and examines and classifies the traffic based on fuzzy inference rules. The anomaly prediction is determined by fuzzy if-then rules which are previously defined. The input variables *Connect* and *ConnectAck* determine the fuzzy variable anomaly by applying the fuzzy rules.

5.3 Fuzzy inference system

The fuzzy inference engine computes the anomaly computation of each client using fuzzy rules. Figure 8 depicts the graphical representation of the proposed fuzzy inference engine. The CMR and CAMR are the input parameters of the proposed scheme, and the output variable anomaly is also a fuzzy variable.

If any anomaly is found, then the broker will discard the packets. Otherwise, the normal packets are sent to the subscriber. Based on the input variables CMR and CAMR, the fuzzy inference engine generates the fuzzy rules in the form of IF-THEN statement. An example of the rule is given as follows:

1. IF CMR = Low and CAMR = Low THEN anomaly = Normal
2. IF CMR = Low and CAMR = Medium THEN anomaly = Abnormal
3. IF CMR = High and CAMR = Medium THEN anomaly = Attack

Table 2 gives the fuzzy rules generated from heuristic fuzzy rule generation method. The proposed model considered all possible permutations of the membership

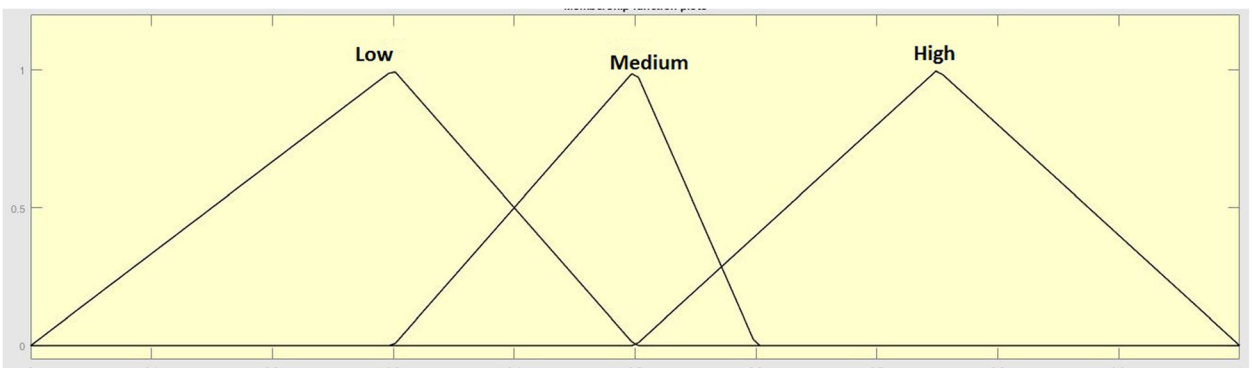

Fig. 6 Membership function of CAMR

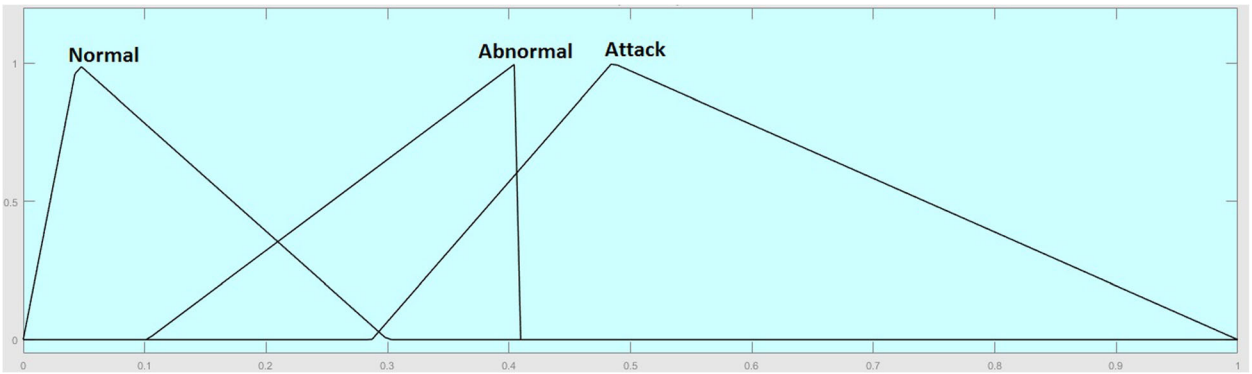

Fig. 7 Membership function of *anomaly*

values of CMR and CAMR. Each permutation is used as an antecedent for each rule. However, few permutations have shown poor performance as it was difficult to infer the consequent. Therefore, to improve the performance, Secure-MQTT adopted fuzzy rule interpolation.

5.4 Fuzzy rule interpolation

Fuzzy inference engine predicts the severity of anomaly by matching the rule found in rule base in the MQTT broker during the communication. It is a challenging task to form an exhaustive set of all the possible rules from the network trace set. A fuzzy inference engine fails to respond if a matching antecedent is not found in the rule base, which results in the poor performance of the fuzzy inference engine. Fuzzy rule interpolation, which executes if the fuzzy inference engine could not find any suitable match in the rule base, can be applied to improve the performance of the IDS. Fuzzy rule interpolation forms new rules based on past scenarios and modifies the rule base dynamically. This dynamic update of the rule base decreases the false-positive and false-negative rates and increases efficiency.

Secure-MQTT follows transformation-based fuzzy rule interpolation [24]. The original rule base R contains the rule R_i and the observation O such that $R_i \in R$. Each rule R_i can be represented as $R_i =$ IF $(x_1 = A_{i1})$ *and* $\ldots (x_j = A_{ij}) \ldots (x_n = A_{iN})$ THEN $y = B_i$ where A_{ij} is the triangular membership functions, x_j is the j^{th} antecedent, $1 \leq j \leq N$, N is the total number of antecedents, and B_i is the consequent. The observation O can be represented as $O = \{A_{o1}, A_{o2}, \ldots, A_{oN}\}$ and $A_{ij} = (l, n, r)$, where l and r are left and right points in the triangle and n is the normal point in the triangular and denotes the triangular membership function of the antecedent x_j. The representative value of triangular function A can be defined as the mean of x-coordinates which is computed as

$$\mathrm{REP}(A) = \frac{(l + r + n)}{3} \tag{3}$$

The steps in transformation-based fuzzy rule interpolation are given below:

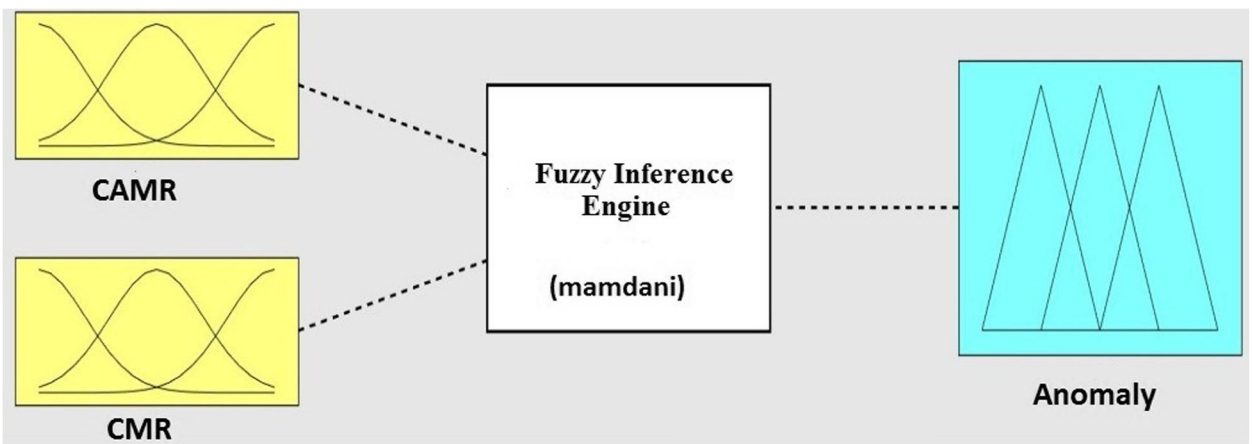

Fig. 8 Proposed Fuzzy Inference Engine

Table 2 Fuzzy rules for the proposed system

CMR	Low	Low	Low	Medium	Medium	Medium	High	High	High
CAMR	Low	Medium	High	Low	Medium	High	Low	Medium	High
Anomaly	Normal	Abnormal	Attack	Normal	Normal	Abnormal	Attack	Attack	Attack

- *Find the nearest rule for new observation:* The aggregate distance of all x_j determines the distance between R_i and the observation O and it is calculated as follows:

$$\mathrm{dist}(R_i, O) = \sqrt{\sum_{j=1}^{N} \mathrm{dist}_j} \text{ and } \mathrm{dist}_j = \frac{\mathrm{dist}(A_{ij}, A_{oj})}{\mathrm{Range}_{x_j}} \quad (4)$$

  where $\mathrm{dist}(A_{ij},A_{oj})$ gives the distance between A_{ij} and A_{oj}, with $Range_{xj}$ for j^{th} antecedent. Then, P rules, $M \geq 2$ with respect to observed value A_{oj}, are selected for interpolation operation to achieve the conclusion C_o.
- *Design the transitional rules:* The approximate value of the final consequent can be derived from the transitional rules using the new observations. This is achieved by applying linear interpolation to the P rules identified. The expected antecedents of the new rule are computed using the antecedents of P rules as follows:

$$A_j' = \sum_{i=1}^{P} W_{ij} A_{ij} \quad (5)$$

  where $W_{ij} = \frac{W'_{ij}}{\sum_{k=1}^{P} W'_{ij}}$, $W'_{ij} = \exp^{-d(Aij - Aoj)}$

  Then, A_j' is mapped to $A_j'' = A_j' + \beta_j Range_{xj}$.

  where β_j is the difference between A_{oj} and A''_j and is calculated as:

$$\beta_j = \frac{\mathrm{REP}(A_{oj}) - \mathrm{REP}(A_j'')}{\mathrm{Range}_{xj}} \quad (6)$$

Also, the aggregated values of W_{Bi} and β_B are computed as follows:

$$W_{Bi} = \frac{1}{N}\sum_{j=1}^{N} W_{ij}, \quad \beta_B = \frac{1}{N}\sum_{j=1}^{N} \beta_j \quad (7)$$

Using this, the mapped consequent B'' can be computed for the antecedents A_j''

- Scaling and moving transformations: The REP values of antecedents of a transitory rule are matches with those of observation. Also, we have to make sure that the fuzzy values in the transitory rule should be the same as the observation scale and move transformation.

  The scaled value (l^*,r^*) is determined such that $r^* - l^* = \delta_j(r'' - l'')$

$$s_j = \frac{r^* - l^*}{r' - l'} \quad (8)$$

  Similarly, the consequent's scaling factor is computed as follows:

$$\delta B = \frac{j\sum_{j=1}^{N} \delta j}{N} \quad (9)$$

  The function *move* is applied to the resulting fuzzy values if the mapped fuzzy set is the same as that of observation's and is defined as follows:

$$\begin{cases} \mathrm{move}_j = \frac{3(l - l^*)}{n^* - l^*} & , n \geq n^* \\ \mathrm{move}_j = \frac{3(l - l^*)}{r^* - n^*} & , \text{Otherwise} \end{cases}$$

  From the above equation, the *move* function of the consequent can be derived as follows:

$$\mathrm{move}_B = \frac{\sum_{j=1}^{N} \mathrm{move}_j}{N}$$

  The scale and move mapping are applied to B'' using δ_B and move_B in order to obtain B_o.

5.5 Defuzzification

The crisp output is obtained by using a Mamdani fuzzy inferencing engine [25], which aggregates the fuzzy rules. It follows centroid approach in which the center of the area under the curve of a membership function gives the crisp output.

5.6 Proposed anomaly detection algorithm

The proposed anomaly detection algorithm is performed by MQTT broker and the procedures of the algorithm are formally stated in Algorithm 2. The proposed algorithm 2 takes input as the MQTT packet flowing through the network and returns the decision whether to accept or reject the packet as output.

Algorithm 2 Detection of DoS attack in MQTT

Input: MQTT packet flowing through the network

Output: Decision whether to accept or drop the packet

```
BEGIN
    Anomaly ← NULL
    Examine the arrival time of MQTT packets from the publishers
    Select the relevant features, Connect and ConnAck
    Compute CMR = N_Connect / N
    Compute CAMR = N_ConnAck / N
    Fuzzify CMR and CAMR
    Apply Fuzzy Inference Rule Engine
    IF rule found THEN
        Defuzzify the Anomaly
    ELSE
        Apply Fuzzy Rule Interpolation
    IF Anomaly == attack THEN
        Decision ← Drop the packet
    ELSE
        Decision ← Accept the packet
RETURN Decision
END
```

The relevant features *Connect* and *ConnAck* are selected from the observed network traffic, and the fuzzy variables CMR and CAMR are determined by Eqs. 1 and 2. Both CMR and CAMR are fuzzified and fuzzy inference engine, which selects the appropriate rule, is applied. If such a rule is not found in the rule base, then fuzzy rule interpolation is executed. Then, defuzzify the output variable anomaly and the decision is made. The decision is either packet accept or drop as per the anomaly.

6 Results and discussion

The Secure-MQTT is implemented in Contiki OS and the evaluation is done using the Contiki simulator COOJA. We run emulations on the sensor platform T1 EXP 5438 having a 16-bit processor MSP430F5438A with 256 KB flash, and 16 KB RAM MSP430F5438A has a 25-MHz clock frequency. The area of the IoT network is 500 × 500 m^2 in which 60 to 500 devices are deployed. The IoT network is modeled with legitimate nodes and the attacker nodes in the simulation environment. The attack is created in a distributive nature in which the number of attacker nodes varies from 10% to 50% of the total number of nodes present in the network. The attacking node sends the same request frequently at a higher rate compared to a legitimate node. We compare Secure-MQTT with MQTT-S [26], and the comparison results demonstrate that Secure-MQTT outperforms MQTT-S.

6.1 IDS evaluation metrics

The proposed scheme employs the significant IDS metrics such as attack detection efficiency, attack detection accuracy ratio, attack detection rate, false-positive ratio, and precision rate to verify and validate the performance. The experiment is simulated with a varying total number of nodes deployed in the network. Precisely, we considered four scenarios, 100, 150, 200, and 300 nodes out of which 10% of nodes were simulated as malicious nodes. The simulation repeats over different time frames T1, T2, T3, and T4 and the overall performance is evaluated. Each time frame represents a fixed duration with varying publish-subscribe messages. We assume that these messages follow uniform distribution for legitimate nodes whereas the malicious nodes show sharp variation from the normal flow of messages.

6.1.1 Attack detection efficiency (ADE)

ADE gives the efficiency of the proposed scheme in determining the malicious nodes based on the total number of nodes deployed in the network. The following equation calculates the efficiency of detection.

$$\mathrm{ADE} = \frac{C_{\mathrm{M}}}{N} \tag{10}$$

where C_{M} denotes the number of detected malicious nodes and N represents the total number of nodes present in the network. Figure 3 shows the ADE of Secure-MQTT.

Each deployment scenario (100, 150, 200, 300 nodes) is evaluated separately and the results are given in Fig. 9. In all four cases, Secure- MQTT achieves better ADE compared to MQTT-S as the proposed algorithm captures immediate variation in the message flow. The Secure-MQTT detects the presence of the malicious node by counting CONNECT and CONNACK messages from the device. Hence, Secure-MQTT performs the early detection of the malicious node which attacks a broker through flooding. Also, it is clear from Fig. 9 that the Secure-MQTT shows steady performance in detecting malicious nodes on an average of 80% and above compared to MQTT-S. MQTT-S follows traditional detection procedure using SSL/TLS security. This approach is not suitable for a dynamic network environment like IoT network since the SSL/TLS considers general network traffic features.

6.1.2 Attack detection rate (ADR)

ADR is the number of true positives successfully detected out of the total number of detections and is determined by the following:

$$ADR = \frac{N_{TP}}{(N_{TP} + N_{FN})} \tag{11}$$

Figure 10 shows the ADR of Secure-MQTT in different simulation scenarios.

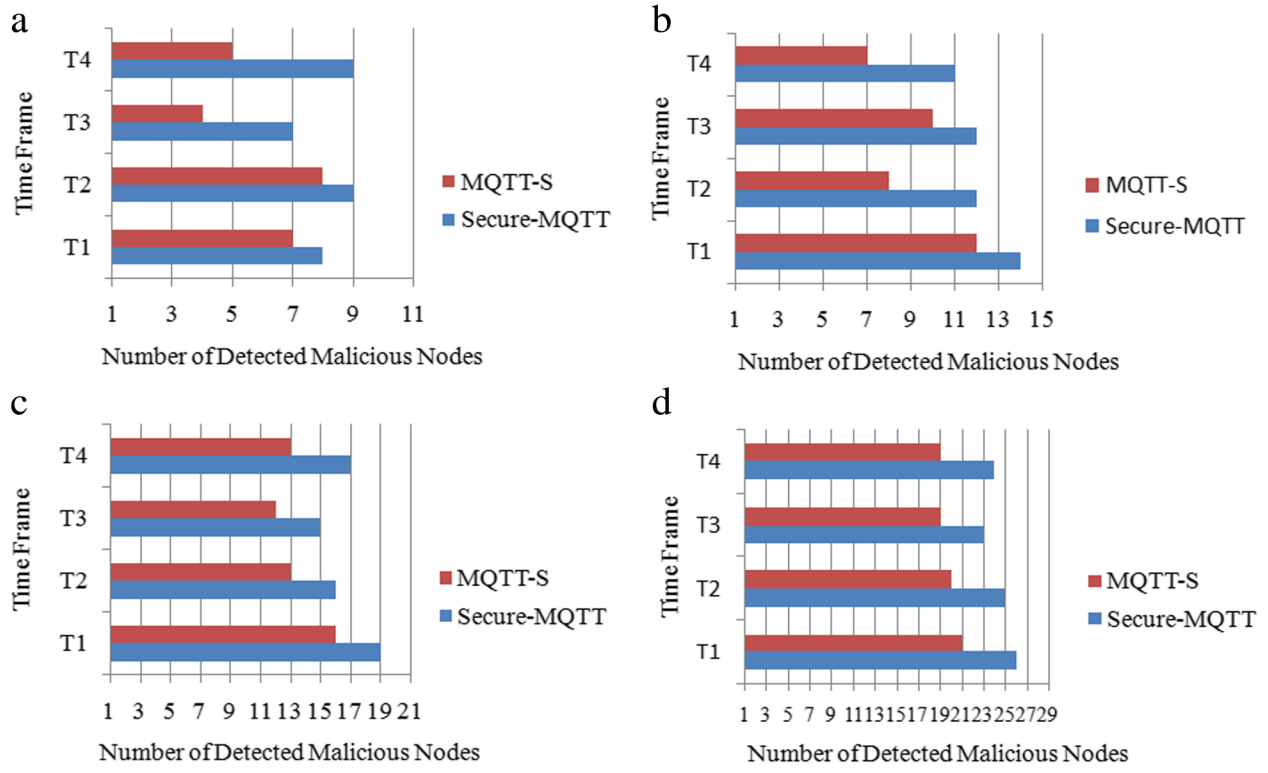

Fig. 9 Attack detection efficiency of **a** 100, **b** 150, **c** 200, and **d** 300 nodes

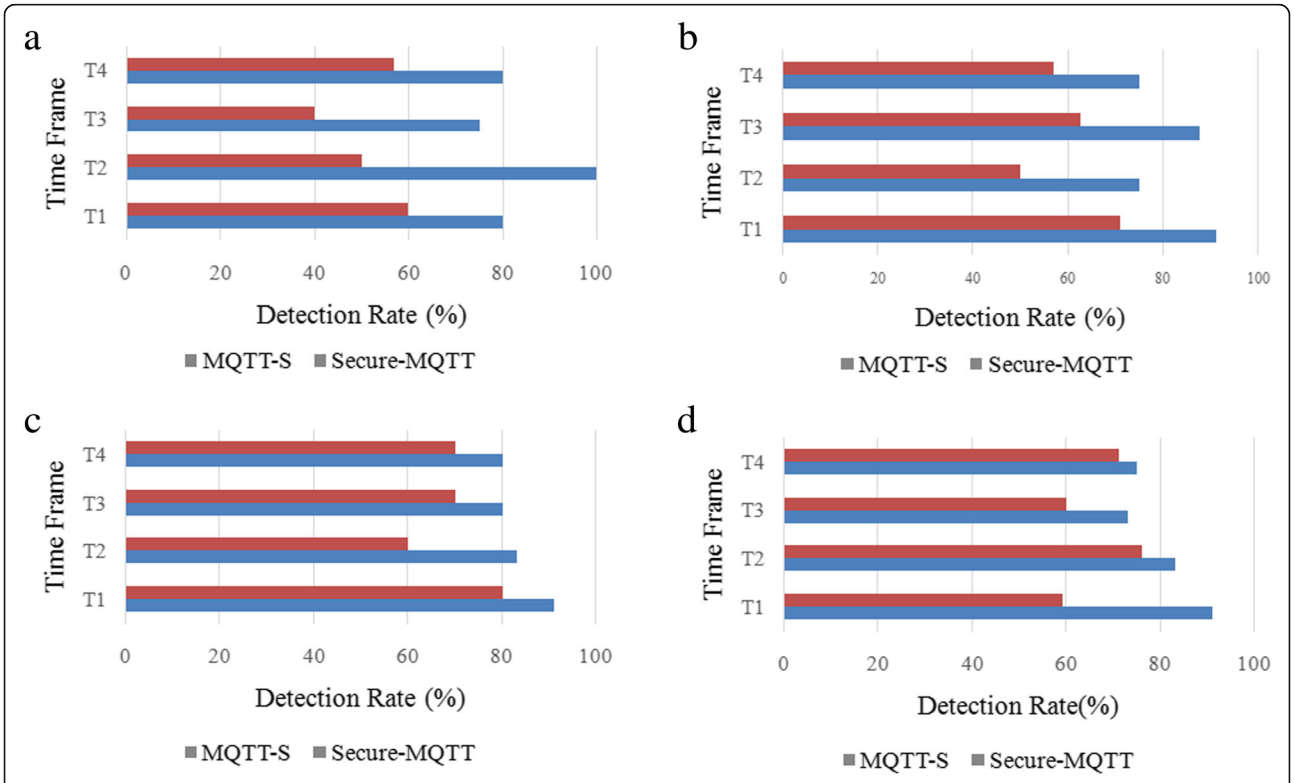

Fig. 10 **a**–**d** Attack detection rate in different scenarios

From Fig. 10, it is observed that Secure-MQTT maintains a steady detection ratio in all the four simulation scenarios as compared to MQTT-S. As discussed in ADA, relevant feature selection helps to create a better rule base. This increases the detection ratio of Secure-MQTT.

6.1.3 Attack detection accuracy (ADA) ratio

ADA ratio is the percentage of malicious nodes detected successfully and is determined by the following formula:

$$\mathrm{ADA} = \frac{N_{\mathrm{TP}} + N_{\mathrm{TN}}}{(N_{\mathrm{TP}} + N_{\mathrm{TN}} + N_{\mathrm{FP}} + N_{\mathrm{FN}})} \tag{12}$$

where N_{TP} is the number of true positives, N_{TN} is the number of true negatives, N_{FP} is the number of false positive, and N_{FN} is the number of false negatives.

Figure 11 shows ADR vs number of devices in Secure-MQTT. The proposed Secure-MQTT achieves high detection accuracy for different network scenarios as compared to MQTT-S. The Secure-MQTT considers only the relevant network traffic features (shown in Table 1) for the decision-making whereas MQTT-S does not employ the feature selection method. This contributes to the better detection accuracy of Secure-MQTT.

6.1.4 False-positive ratio (FPR)

False-positive ratio gives the rate of legitimate nodes that are identified as abnormal nodes and is defined as follows:

$$\mathrm{FPR} = \frac{N_{\mathrm{FP}}}{N_{\mathrm{FP}} + N_{\mathrm{TN}}} \tag{13}$$

Figure 12 shows the FPR of the proposed Secure-MQTT and the MQTT-S over different timeslots by varying the number of request messages. Secure-MQTT obtains the least threshold FPR range [0.1–0.37] as compared to the range [0.3–0.59] of MQTT-S. The fuzzy logic inference engine in Secure-MQTT chooses the most appropriate rules for the decision-making which results in better FPR for Secure-MQTT. The absence of early detection mechanism in MQTT-S increases the FPR.

6.1.4.1 Communications rate The communication rate is defined as the number of bytes published per second from a node, and we have measured the achievable communication rate to show the security processing. It is difficult to measure the computational availability as a metric of the impact of security because the multiprocessing in a sensing device is limited. Thus, it is important to consider the impact of the attack on the maximum achievable communication rate. The total available bandwidth that is assumed is 25,000 B/s for MQTT for the simulation.

Figure 13 describes the maximum communication rate, with and without including IDS to the MQTT protocol. It is observed from Fig. 13 that the proposed Secure-MQTT does not compromise to higher communication rate though it reacts to attacks immediately. This shows that Secure-MQTT is efficient in detecting attacks without compromising network performance.

Precision

Precision gives the number of malicious nodes correctly identified among the detected malicious nodes, which is defined as follows:

$$\mathrm{Precision} = \frac{N_{\mathrm{TP}}}{N_{\mathrm{TP}} + N_{\mathrm{FP}}} \tag{14}$$

We have measured the precision ratio of both the schemes, proposed Secure-MQTT and MQTT-S. Figure 14

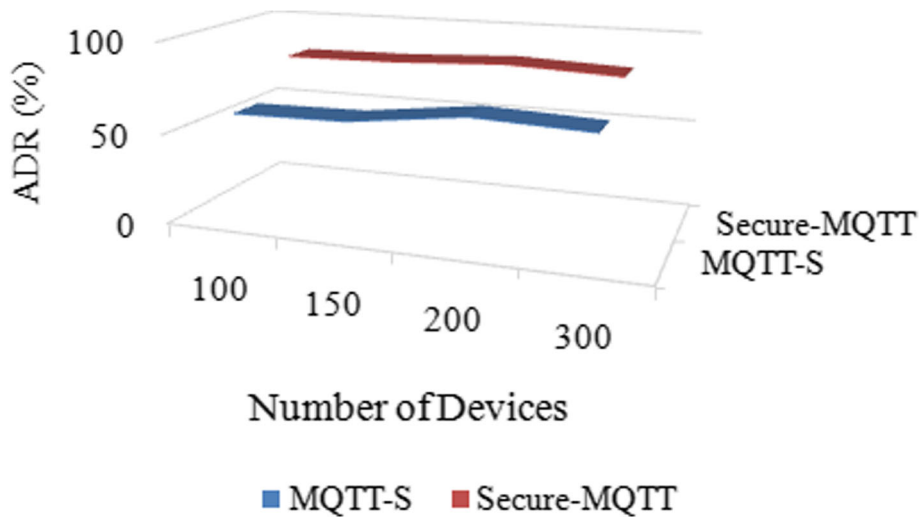

Fig. 11 ADR vs Number of devices in Secure-MQTT

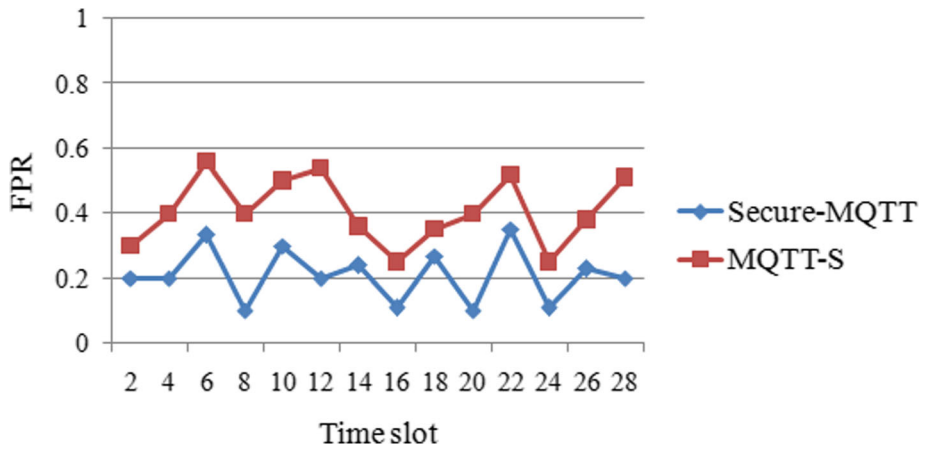

Fig. 12 FPR in different time slots

shows that Secure-MQTT achieves better precision compared to that of MQTT-S.

Secure-MQTT achieves high precision due to the relevant feature selection and fuzzy base inference engine. In Secure-MQTT, the use of fuzzy interpolation enables the fuzzy inference engine to detect almost all possible attacks effectively and leads to a better true-positive rate than MQTT-S. There is no early detection in MQTT-S since it follows the traditional detection method SSL/TLS. The recall and F-score for the Secure-MQTT are calculated as given below:

Recall

Recall gives the percentage of malicious nodes correctly identified among the total detected malicious nodes, which is defined as follows:

$$\text{Recall} = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FN}}} \tag{15}$$

F-score

Harmonic mean of precision and recall gives the F-score and is defined as follows:

$$F\text{–score} = 2.\frac{\text{Precision.Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

The overall performance analysis of the proposed Secure-MQTT is given in Table 3. Here, we consider precision, recall, and F-score for different scenarios in which the network is having 300 nodes with 10% malicious nodes.

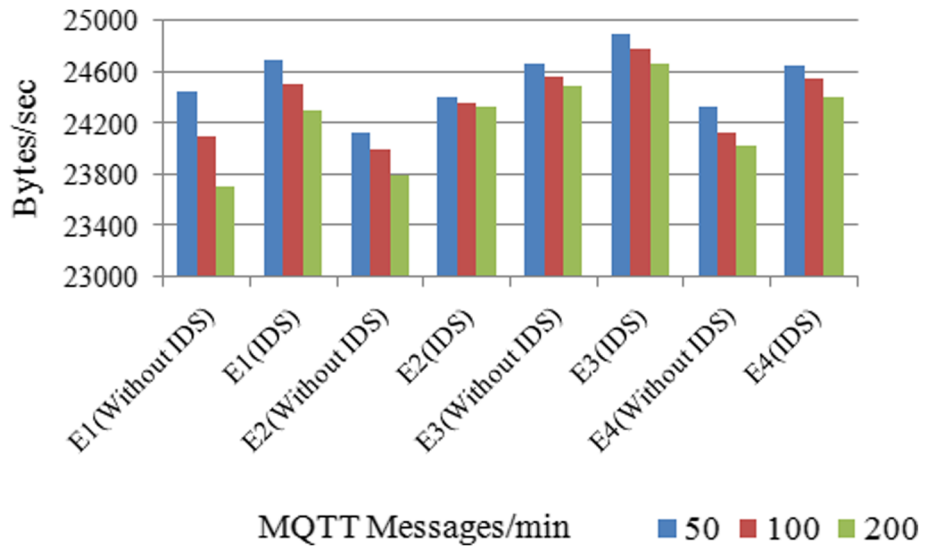

Fig. 13 Communication rate in Secure-MQTT

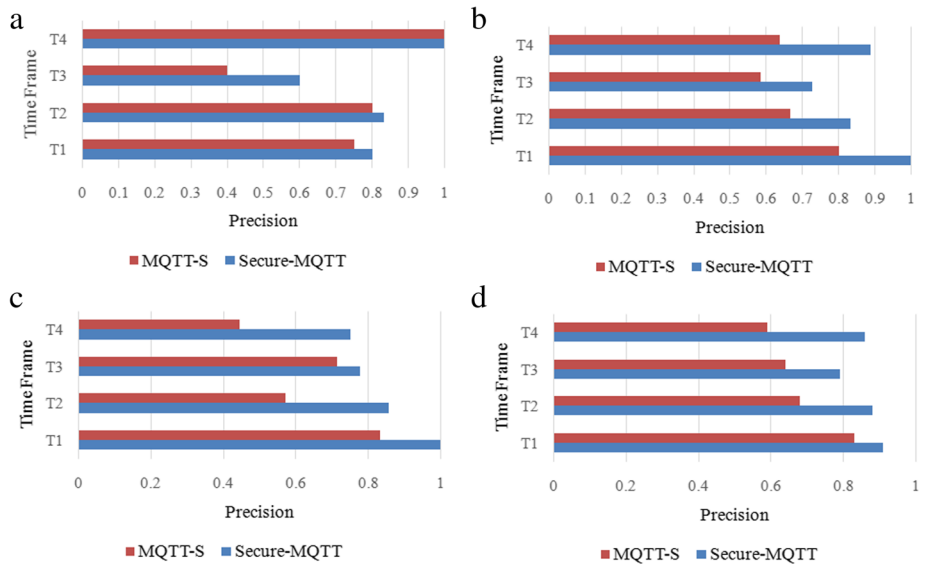

Fig. 14 a–d Precision in different time frames

Secure-MQTT maintains better precision, recall, and F-score as the fuzzy rule interpolation detects the anomaly by generating new rules in the absence of matching rules in the rule base. Table 3 shows the performance analysis of Secure-MQTT under various scenarios. The inference is that the proposed system maintains consistent performance under all possible scenarios. This indicates the significance of Secure-MQTT in real-time applications. Since the proposed Secure-MQTT is a direct application of fuzzy logic, it needs only simple circuit elements. Hence, the computation required for Secure-MQTT is less when compared to existing MQTT-S. Moreover, the proposed Secure-MQTT shows high precision because of the use of fuzzy logic. Fuzzified input and output parameter ensures that there will not be any performance degradation in Secure-MQTT.

7 Conclusions and future work

In this paper, we proposed a novel IDS, Secure-MQTT, for MQTT-based IoT environments. This method uses the correlation-based network feature selection, which selects only relevant features. A fuzzy logic-based inference engine in Secure-MQTT determines the presence of a malicious device accurately. Fuzzy rule interpolation makes the Secure-MQTT lightweight. There is no need to store all the rules in the rule base. Fuzzy rule interpolation dynamically forms the rules, if the matching antecedent is not found in fuzzy inference engine. Hence, the storage of exhaustive sets of rules is avoided in Secure-MQTT and it is computationally inexpensive to have a match checking run for each instance. Moreover, the fuzzy-based Secure-MQTT needs only simple circuit elements for system implementation. This also leads to the design of a lightweight Secure-MQTT by saving computation time and energy. The simulation of different attack scenarios shows the effectiveness of Secure-MQTT in detecting malicious in IoT. The experimental analysis shows that Secure-MQTT achieves low FPR compared to existing work.

This work can be extended by considering other MQTT messages like SUBACK, PUBLISH, and PUBREC in order to strengthen the application layer security. Also, based on observed flexibility and scalability of the Secure-MQTT, this work is extensible to detect various attacks in other communication layers in IoT. As a future work, the proposed Secure-MQTT can be modified with the optimization of selected network traffic features.

Table 3 Performance analysis of Secure-MQTT

Scenario	Positive	False negative	False positive	Precision	Recall	F-score
1	20	2	2	0.9090	0.9090	0.9090
2	15	3	2	0.8823	0.8333	0.8571
3	11	4	3	0.7857	0.7333	0.7586
4	12	4	2	0.8571	0.75	0.80

Acknowledgements
Authors thanks in advance all the reviewers for their valuable suggestions and comments

Funding
Not Applicable

Authors' contributions
HAP proposed the idea, completed the simulation, and analyzed the performance of the proposed work. KK analyzed the factors that influence the algorithms and gave valuable suggestions to improve the manuscript. Both authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 30 May 2018 Accepted: 17 March 2019
Published online: 05 April 2019

References

1. L. Atzori, A. Iera, G. Morabito, The Internet of Things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
2. P. Sethi, S.R. Sarangi, Internet of Things: architectures, protocols, and applications. Can. J. Electr. Comput. Eng. **2017**(2017), 1–25 (2017)
3. C. Gomez, A. Arcia-Moret, J. Crowcroft, TCP in the Internet of Things: from ostracism to prominence. IEEE Internet Comput. **22**(1), 29–41 (2018)
4. H. Lampesberger, Technologies for web and cloud service interaction: a survey. SOCA **10**(2), 71–110 (2016)
5. S. Vinoski, Advanced message queuing protocol. IEEE Internet Comput. **10**(6), 87–89 (2006)
6. M. Ammar, G. Russello, B. Crispo, Internet of Things: A survey on the security of IoT frameworks. J. Inf. Secur. Appl. **38**, 8–27 (2018)
7. R.A. Light, Mosquitto: server and client implementation of the MQTT protocol. J. Open Source Softw. **2**(13), 265 (2017)
8. S. Shadroo, A.M. Rahmani, Systematic survey of big data and data mining in internet of things. Comput. Netw. **139**, 19–47 (2018)
9. S. Jang, D. Lim, J. Kang, I. Joe, An efficient device authentication protocol without certification authority for Internet of Things. Wirel. Pers. Commun. **91**(4), 1681–1695 (2016)
10. A. Oyler, H. Saiedian, Security in automotive telematics: a survey of threats and risk mitigation strategies to counter the existing and emerging attack vectors. Secur. Commun. Netw. **9**(17), 4330–4340 (2016)
11. S. Pang, D. Komosny, L. Zhu, R. Zhang, A. Sarrafzadeh, T. Ban, D. Inoue, Malicious events grouping via behavior based darknet traffic flow analysis. Wirel. Pers. Commun. **96**(4), 5335–5353 (2017)
12. B.B. Zarpelão, R.S. Miani, C.T. Kawakani, S.C. de Alvarenga, A survey of intrusion rjr. J. Netw. Comput. Appl. **84**, 25–37 (2017)
13. M. Roesch, Snort: Lightweight intrusion detection for networks. Lisa **99**(1), 229–238 (1999)
14. D. Oh, D. Kim, W.W. Ro, A malicious pattern detection engine for embedded security systems in the Internet of Things. Sensors **14**(12), 24188–24211 (2014)
15. E. Cho, J. Kim, C. Hong, *Attack model and detection scheme for botnet on 6LoWPAN Management Enabling the Future Internet for Changing Business and New Computing Services, Lecture Notes in Computer Science 5787* (Springer, Berlin, Heidelberg, 2009), pp. 515–518
16. R. Stephen, L. Arockiam, Intrusion detection system to detect sinkhole attack on RPL protocol in Internet of Things. Int. J. Electr. Electron. Comput. Sci. **4**(4), 16–20 (2017)
17. S. Shamshirband, A. Patel, N.B. Anuar, M.L.M. Kiah, A. Abraham, Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks. Eng. Appl. Artif. Intell. **32**, 228–241 (2014)
18. D. Lavrova, A. Pechenkin, Applying correlation and regression analysis to detect security incidents in the internet of things. Int. J. Commun. Netw. Inf. Secur. **7**(3), 131 (2015)
19. S. Raza, L. Wallgren, T. Voigt, SVELTE: Real-time intrusion detection in the Internet of Things. Ad Hoc Netw. **11**(8), 2661–2674 (2013)
20. L. Wallgren, S. Raza, T. Voigt, Routing attacks and countermeasures in the RPL-based internet of things. Int. J. Distrib. Sens. Netw. **9**(8), 794326 (2013)
21. M.A. Prada, P. Reguera, S. Alonso, A. Morán, J.J. Fuertes, M. Domínguez, Communication with resource-constrained devices through MQTT for control education. IFAC-PapersOnLine **49**(6), 150–155 (2016)
22. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutorials **17**(4), 2347–2376 (2015)
23. L. Yang, Q. Shen, Adaptive fuzzy interpolation. IEEE Trans. Fuzzy Syst. **19**(6), 1107–1126 (2011)
24. Z. Huang, Q. Shen, Fuzzy interpolative reasoning via scale and move transformations. IEEE Trans. Fuzzy Syst. **14**(2), 340–359 (2006)
25. E. Pourjavad, A. Shahin, The application of Mamdani fuzzy inference system in evaluating green supply chain management performance. Int. J. Fuzzy Syst. **20**(3), 901–912 (2018)
26. U. Hunkeler, H.L. Truong, A. Stanford-Clark, in *3rd International Conference on IEEE Communication Systems Software and Middleware and Workshops, 2008.* MQTT-S A publish/subscribe protocol for Wireless Sensor Networks (2008), pp. 791–798