Register
Scientific Journal of Information Systems Technology (Scientific Journal of Information System Technology)
ISSN 2502-3357 (Online)
ISSN 2503-0477 (Print)
Available online to www.journal.unipdu.ac.id
Unipdu
S2-Accredited - Decree No. 34 / E / KPT / 2018
Journal Page is available towww.journal.unipdu.ac.id:8080/index.php/register

Advanced Detection of Denial of Service Attack in the Internet of Things Network Based on MQTT Protocol Using Fuzzy Logic

Mochamad Soebagja Budiana [a], Ridha Muldina Negara [b], Arif Indra Irawan [c], Harashta Tatimma Larasati [de]

[a,b,c] *School of Electrical Engineering, Telkom University, Bandung, Indonesia*
[d] *School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung 40116, Indonesia*
[e] *School of Computer Science and Engineering, Pusan National University, Busan 609735, Korea*

email:[a] mochsoebagja@gmail.com, [b] ridhanegara@telkomuniversity.ac.id, [c] arifirawan@telkomuniversity.ac.id, [de] harashta@pusan.ac.kr

ARTICLEINFO

Article history:
Received xxx Revised xxx Accepted xxx Available online xxx

Keywords: [Key word heading]
fuzzy-logic
message queuing telemetry transport (MQTT)
internet of things (IoT)
denial of service (DoS)

IEEE style in citing this article: [citation Heading]
MS Budiana, RM Negara, and AI Irawan, "Advanced detection denial of service attack in internet of things network based on MQTT protocol using fuzzy logic," Register: Scientific Journal of Information Systems Technology, vol. xxx, no. xxx, pp. xxx, 2021.

ABSTRACT

Message Queuing Telemetry Transport (MQTT) is one of the popular protocols used on the Internet of Things (IoT) networks because of its lightweight nature. With the increasing number of devices connected to the internet, there is a growing risk of cybercrimes on the IoT networks. One of the most popular attacks is the Denial of Service (DoS) attack. Standard security on MQTT uses SSL/TLS, but SSL/TLS is computationally wasteful for low-powered devices. On the other hand, fuzzy logic algorithms with the Intrusion Detection System (IDS) scheme is more suitable for detecting DoS because of their simple nature. In this paper, we employ a fuzzy logic algorithm embedded in a node to detect DoS for the MQTT protocol with feature selection nodes. This paper's contribution is that the nodes feature selection used will monitor SUBSCRIBE and SUBACK traffic and provide this information to fuzzy input nodes to detect DoS attacks. Fuzzy performance evaluation is measured against changes in the number of nodes and attack intervals. The results obtained are that the more the number of nodes and the higher the traffic intensity, the fuzzy performance will decrease, and vice versa. However, the number of nodes and traffic intensity will affect the decrease in the detection parameters of fuzzy DoS attacks.

1. Introduction

The Internet of Things (IoT) network or Machine-to-Machine (M2M) communication has become an essential part of today's era[1]. An IoT network connects things or objects (such as devices, cars, or sensors) with each other via wireline or wireless media to the internet network. IoT's primary purpose is to connect all entities anywhere and anytime[2]. In general, entities on an IoT network have limited resources (such as memory, storage, or power)[3]. IoT devices may have an 8-bit microcontroller specification with only 20kB of RAM and 100kB of ROM[2]. For the device to be able to connect to the internet, addressing using internet protocol (IP) is required, which can be addressed by leveraging IPv6 over Low power Wireless Personal Protocol (6LoWPAN)[4]. Several other protocols that support IoT networks are Constrained Application Protocol (CoAP), Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP), and Message Queuing Telemetry Protocol (MQTT)[1][5][6].

ISSN 2502-3357 (Online)
ISSN 2503-0477 (Print)

The MQTT protocol has a light and simple characteristics. It is one of the best candidates for use on networks with constrained, low-bandwidth, high-latency, or unreliable network[7]. MQTT uses a publish/subscribe communication pattern[8], which has three main elements: publisher, subscriber, and broker. Publishers (e.g., sensors, PCs, smartphones) provide data and publish topics to brokers. Subscribers (e.g., applications or devices) are the party who requests the topic from the brokers. The broker acts as a server and is responsible for exchanging topics between publishers and subscribers[9]. However, the light and simple characteristics of the MQTT may cause safety issues[10]; hence MQTT uses SSL / TLS as its standard security to prevent eavesdropping attacks or prevent damage to the data. Unfortunately, SSL / TLS is not designed to handle DoS attacks[11]. One of the methods to avoid DoS attacks is to use an Intrusion Detection System (IDS)[9]. In the implementation, IDS can utilize fuzzy logic[1], machine learning[12], deep learning[13], or even blockchain[5]. Regarding the fuzzy logic algorithm, applying it to the MQTT protocol is a suitable choice because it has several advantages. First, it has a good approach to decision-making problems. Second, it employs easy and simple implementation[14].

The fuzzy logic algorithm was invented by Lotfy Zadeh in 1965 [15][16], which was initially studied by Lukasiewicz in 1920 as many-valued logic [17]. Fuzzy logic is a development of boolean logic[18], in which fuzzy logic is based on "human reasoning" in order to get an output that is close to "true" [19]. The approach using fuzzy logic will provide a comprehensive output than using boolean logic[20]. In boolean logic, there are only two truth values, namely "true" (usually 1) or "false" (usually 0), so the output is "completely true" or "completely false". In contrast, fuzzy logic covers a broader degree of truth, ranging from 0 to 1, thus producing the outcome in the form of "partially true" or "partially false"[18]. There are four main components in the fuzzy logic algorithm: fuzzification, fuzzy rules, fuzzy inference engine, and defuzzification[14]. Input data must pass through a fuzzifier block so that the input data can be converted from numeric form to linguistic form. After the fuzzification stage, the fuzzy inference engine or Fuzzy Inference System (FIS) will determine the output based on linguistic information. This stage uses an approach based on human interpretation. The last stage is the defuzzification block, where the fuzzy output will be converted and translated into the output in the form of decisions to be made[19][20] [21].

This paper discusses fuzzy logic algorithms' performance for detecting DoS attacks on IoT networks based on the MQTT protocol. In our experiment, the algorithm is planted on one of the network nodes as a subscriber and will detect DoS attacks with feature selection nodes. To determine the performance of fuzzy logic, ascenario of changing the number of nodes in the network topology and the number of attack nodes is used.

2. Related Work

The limitations and simplicity of the MQTT pose problems to its security system. This problem makes MQTT vulnerable to a wide variety of attacks[10]. Various kinds of security can be implemented in MQTT, depending on the type of attack. Haripriya et al. in [1] used fuzzy logic-based IDS to identify network anomalies against DoS attacks based on PUBLISH and SUBSCRIBE traffic from each client node. Both traffic is used as input variables by the fuzzy. Fuzzy will then calculate the degrees of the two variables. It executes the logic "IF ... THEN ..." using the fuzzy inference system (FIS). In particular, the FIS method used is the Mamdani method. Additionally, this study uses fuzzy rule interpolation to generate new rules based on network traffic flow behavior. The result of this research is a system capable of detecting DoS attacks on the MQTT protocol.

Harsha et al. [6] discussed the identification of security holes in MQTT using Shodan API based on the MQTT package and the QoS level. This paper shows that most MQTT users do not use authentication mechanisms and data authority, making the network vulnerable to sniffing or data modification. The prevention mechanism proposed by Harsha et al. to improve authentication on MQTT, among others, is by using a username and password between the client and the broker, implementing the plain text in CONNECT messages, using private lines (TLS), or using encryption on the client and decryption at the broker. Furthermore, improving data authority is performed by using ACLs per topic, per method, or QoS. Ivan et al. validated DoS attacks named SlowITe and SlowTT on the MQTT service [21][22] The attack exploited a weakness in the Keep-Alive parameter setting which

ISSN 2502-3357 (Online)

ISSN 2503-0477 (Print)

is used to keep the connection alive by avoiding connection closures by the server. The results indicate that these attacks can exploit vulnerabilities in the MQTT network, regardless of whether communications are encrypted or not.

Andy et al. [7] discussed the handling of sniffing attacks, data modification, and Botnet on MQTT. They propose to use TLS, ECC, or ECC and RSA to handle sniffing attacks and data modification on private networks. Meanwhile, to deal with Botnet attacks (DoS or phishing), the authors propose to put MQTT broker on a public network. In addition, Potrino et al. in [9] presented Host-based IDS (HIDS) attached to fog nodes to handle SYN flooding attacks, CONNECT flooding attacks, High QoS message attacks, and DoS using PUBLISH attacks on MQTT. HIDS identifies and validates the buffer length and QoS level of CONNECT and PUBLISH messages coming from each client. Ramos et al. [21] proposed a framework using template-based fuzzing techniques to detect DoS attacks on MQTT, whereas for this study, we aim to detect DoS attacks on the MQTT protocol using fuzzy logic algorithms. SUBSCRIBE and SUBACK variables are used as the input variables in fuzzy logic. Fuzzy logic will be placed on network nodes and will monitor MQTT traffic flow with feature selection nodes.

3. Research Method

3.1. System Design

In our design, the fuzzy logic algorithm is embedded in one of the nodes, which is called a fuzzy node. Also, five additional nodes are required as support nodes to observe the network's MQTT traffic flow as each node is only capable of subscribing to one topic. Figure 1 shows the block diagram of the whole system. Feature selection nodes will monitor MQTT traffic flow in real-time, where the flow information is obtained from the broker using the broker status topic. There are five kinds of data needed by the feature selection nodes: the number of input packet to the broker, the number of output packet from the broker, the number of PUBLISH input to the broker, the number of PUBLISH output from the broker, the number of CONNECT packet, and the number of CONACK packet.

Furthermore, feature selection nodes will send back all MQTT traffic flow information to the broker using a new topic. Hence, the fuzzy node can find out information about the MQTT traffic flow using topics that the feature selection nodes have created and also filter the number of traffic flow information to obtain the number of SUBSCRIBE and the number of SUBACK packets, the SMR value (using Equation 1), and the SAMR value (using Equation 2). SMR defines the ratio between the number of SUBSCRIBE packets with all input packets that enter the broker, whereas SAMR represents the ratio between the number of SUBACK packets and all the output packets that leave the broker.

$$\text{SMR} = \frac{\text{SUBSCRIBE}}{\text{Input Packet}} \quad (1)$$

$$\text{SAMR} = \frac{\text{SUBACK}}{\text{Output Packet}} \quad (2)$$

Subsequently, SMR and SAMR values will be entered into a fuzzy logic algorithm and then converted into linguistic variables. Furthermore, the fuzzy logic algorithm will adjust the SMR and SAMR linguistic variables with the rule bases that have been created using the logic "IF ... AND ... THEN ...". The results of adjusting SMR and SAMR with rule bases will produce linguistic data about network conditions against DoS attacks.

 http://doi.org/10.26594/register.v6i1.idarticle

ISSN 2502-3357 (Online)
ISSN 2503-0477 (Print)

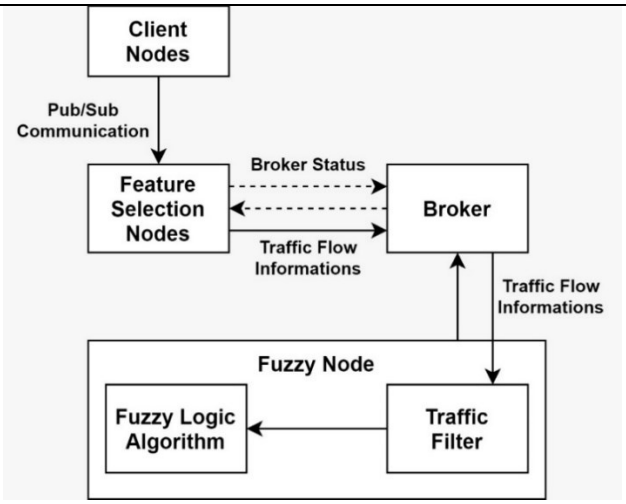

Figure 1. System design

3.2. Feature Selection (FS) Nodes

Nodes Feature Selection (FS) is responsible for observing the MQTT traffic flow on the network. Figure 2 shows a block diagram of the FS nodes. FS nodes will subscribe to the broker regarding traffic flow information using the broker status topic, namely $ SYS / broker / #, so that the broker will publish the required information in real-time. When the FS nodes get information from the broker, the FS nodes will publish the traffic flow information to the broker using a new topic, namely fuzzy / dtc / #, and the broker will save the topic. The required number of FS nodes is five nodes. This condition is due to the device's limited characteristics, in which each node cannot subscribe to more than one topic.

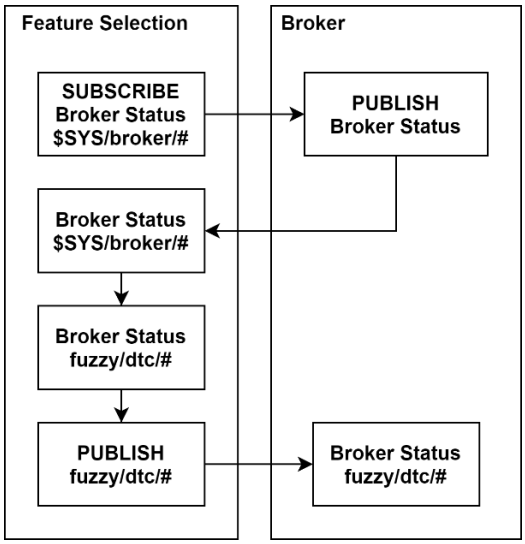

Figure 2. FS model nodes

3.3. Fuzzy Node

Fuzzy nodes are used to detect DoS attacks. Figure 3 shows a block diagram of fuzzy nodes. The fuzzy nodes will subscribe to the broker using the FS node's topics, namely fuzzy /dtc / # so that the fuzzy nodes can find out information about the MQTT traffic flow on the network. After the broker receives a subscribe from the fuzzy node, the broker will publish the fuzzy /dtc / # topic in real-time. Suppose the MQTT traffic flow information from the broker has been received. In that case, the fuzzy node will sort the SUBSCRIBE and SUBACK traffic using Equations 1 and 2 to be used as input variables in the fuzzy logic algorithm. The fuzzy node will then calculate the ratio between SUBSCRIBE packets with the number of input packets using Equation 1 (i.e., SMR) and the ratio between SUBACK packets and the number of output packets using Equation 2 (i.e., SAMR).

http://doi.org/10.26594/register.v6i1.idarticle

ISSN 2502-3357 (Online)

ISSN 2503-0477 (Print)

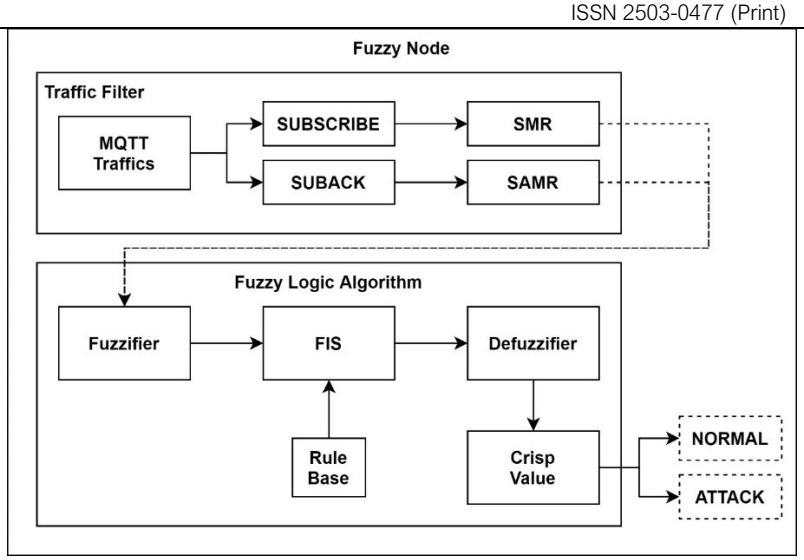

Figure 3. Fuzzy node model

The SMR and SAMR values will be forwarded to the fuzzifier block, where this block will calculate the degree of SMR and SAMR and classifies to the membership function (MF) based on each degree. Each variable will be grouped into linguistic variables based on their respective degrees of MF using the fuzzification process. There are three kinds of linguistic variables in SMR and SAMR, including "LOW," "MEDIUM," and "HIGH". The MF generated by the fuzzifier will be assigned to the FIS block. At this stage, each MF will be adjusted to the rule bases that have been created using the logic "IF ... AND ... THEN ...". Table 1 shows the draft rule bases used. Adjusting MF to rule bases will result in conclusions in linguistic variables that indicate network conditions against DoS attacks. There are three kinds of conclusions produced: "NORMAL" represents the network is safe, "ABNORMAL" indicates that there is suspicious activity on the network, and "ATTACK", which signifies that there is a DoS attack on the network. The conclusions obtained will be converted into crisp value by the defuzzifier block using the defuzzification process.

Table 1. Rule Bases

SMR	L	L	L	M	M	M	H	H	H
SAMR	L	M	H	L	M	H	L	M	H
Output	N	Ab	At	N	N	Ab	At	At	At

3.4. Experiment Setup

The fuzzy control or FIS method used is Mamdani, with the defuzzification method used is the centroid. The simulation utilizes the COOJA network simulator with Mosquitto as the broker platform. The MQTT version used is MQTT v.3.1 with MQTT QoS level 0. The simulation was carried out in various scenarios, namely using 15, 20, 25, and 30 nodes, with each scenario uses an attack interval of 3, 5, 7, and 9 seconds. The number of attack nodes is 20% of the total number of nodes. The attacker will flood the network (flooding attack) using SUBSCRIBE traffic.

4. Results and Analysis

4.1. Membership Function (MF) of Fuzzy Logic Algorithm

Table 2 and Table 3 show the SMR and SAMR values obtained using Equation 3 and Equation 4, respectively. As can be inferred from both tables, the SMR and SAMR values will increase with the increase in the number of nodes or the increase in the attack's intensity. This is because the more the number of nodes, the more the network's traffic will increase. The highest SMR value was obtained in the 25-nodes scenario during the ATK 3S attack interval with a value of 0.6308, while the lowest SMR value was obtained in the 20-nodes scenario when there was no attack with a value of 0.0541. The

http://doi.org/10.26594/register.v6i1.idarticle

ISSN 2502-3357 (Online)
MS Budiana et. al./Register xxx ISSN 2503-0477 (Print)

highest SAMR value was obtained during the ATK 3S attack interval at 15-nodes scenario with a value of 0.3848, while the lowest SAMR value was obtained when there was no attack on 20-nodes with a value of 0.043.

Table 2. SMR value per scenario

Scenarios	15 Nodes	20 Nodes	25 Nodes	30 Nodes
ATK 3S	0.5989	0.5582	0.6308	0.5982
ATK 5S	0.4491	0.4961	0.5740	0.5636
ATK 7S	0.4131	0.4293	0.4652	0.4707
ATK 9S	0.3628	0.3413	0.4405	0.4078
No ATK	0.0724	0.0541	0.1822	0.1092

Table 3. SAMR value per scenario

Scenario	15 Nodes	20 Nodes	25 Nodes	30 Nodes
ATK 3S	0.3848	0.3455	0.3200	0.2410
ATK 5S	0.2951	0.2217	0.2896	0.1421
ATK 7S	0.2446	0.1351	0.2751	0.1988
ATK 9S	0.2698	0.1620	0.1918	0.1233
No ATK	0.0740	0.0430	0.0816	0.1654

Figures 4 and 5 show a graph of the MF SMR and SAMR values between the occurrence of a DoS attack and no attack can be seen clearly so that SMR and SAMR can be used to detect DoS attacks. . The highest average SMR value occurs in the 25-nodes scenario with a value of 0.5276, while the lowest average SMR value occurs in the 20-nodes scenario with a value of 0.0541. The highest average SAMR value occurs in the 15-nodes scenario with a value of 0.2986, while the lowest average SAMR value occurs in the 20-nodes scenario with a value of 0.043. Based on the obtained values shown in Figures 4 and 5, the forms of MF SMR and SAMR used to detect DoS attacks are shown in Figures 6 and 7.

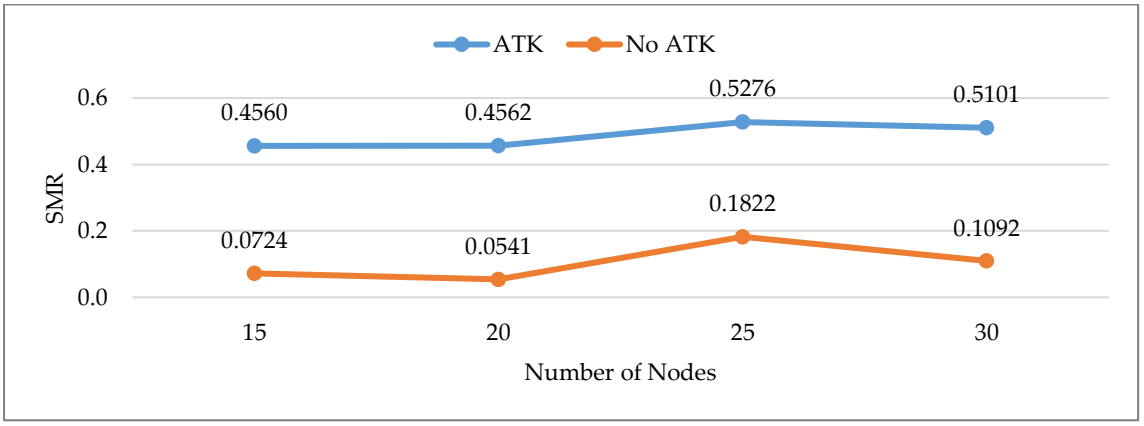

Figure 4. Average of SMR value

ISSN 2502-3357 (Online)
---- et. al./Register xxx ISSN 2503-0477 (Print)

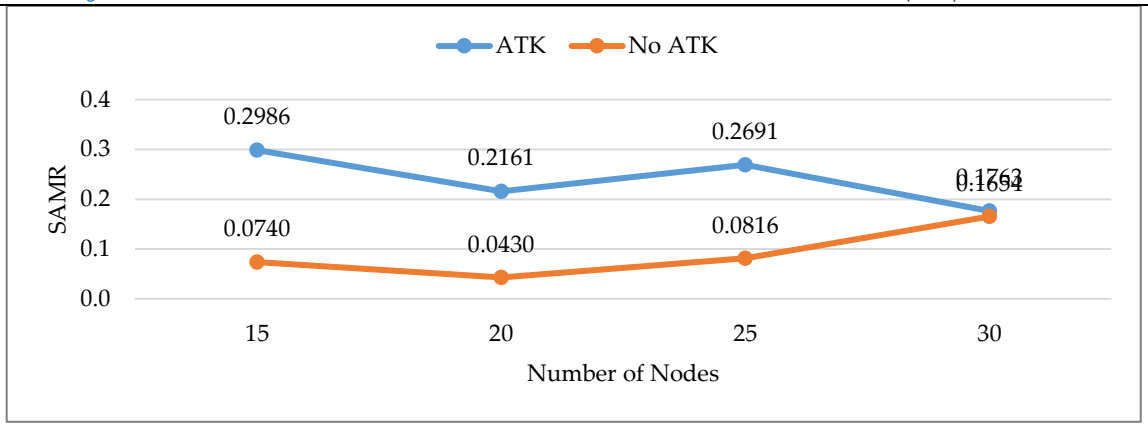

Figure 5. Average of SAMR value

Decision on the form of MF output is obtained based on the experimental results using the rule bases as previously shown in Table 1. There are three MF outputs to indicate network conditions, including "NORMAL," "ABNORMAL," and "ATTACK." "L," "M," and "H" in the SMR and SAMR variables are "LOW," "MEDIUM," and "HIGH." "N," "Ab," and "At" in the output variable are "NORMAL," "ABNORMAL," and "ATTACK." Crisp value is obtained using the defuzzification process. In this study, the defuzzification method used is the centroid method, which means that the crisp value will be obtained based on the resulting MF output's midpoint. To determine the network conditions, (i.e,. whether there is no attack or an attack is in progress), the centroid of MF "ABNORMAL" is used as the reference value where the midpoint of MF "ABNORMAL" is 0.375. Figure 8 is the MF of the output variable, which shows the network condition where if the crisp value falls between zero and the reference value ($0 \leq$ crisps <0.375), it is assumed that there is no DoS attack on the network. In contrast, if the crisp value is greater than or equal to the reference value ($0.375 \leq$ crisps ≤ 1), it is assumed that a DoS attack is taking place on the network.

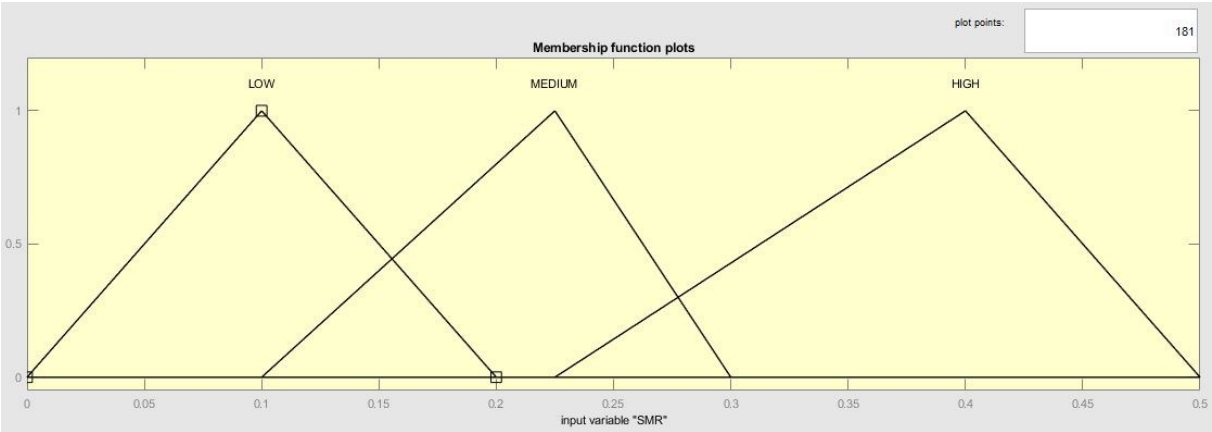

Figure 6. SMR membership function

ISSN 2502-3357 (Online)
MS Budiana et. al./Register xxx ISSN 2503-0477 (Print)

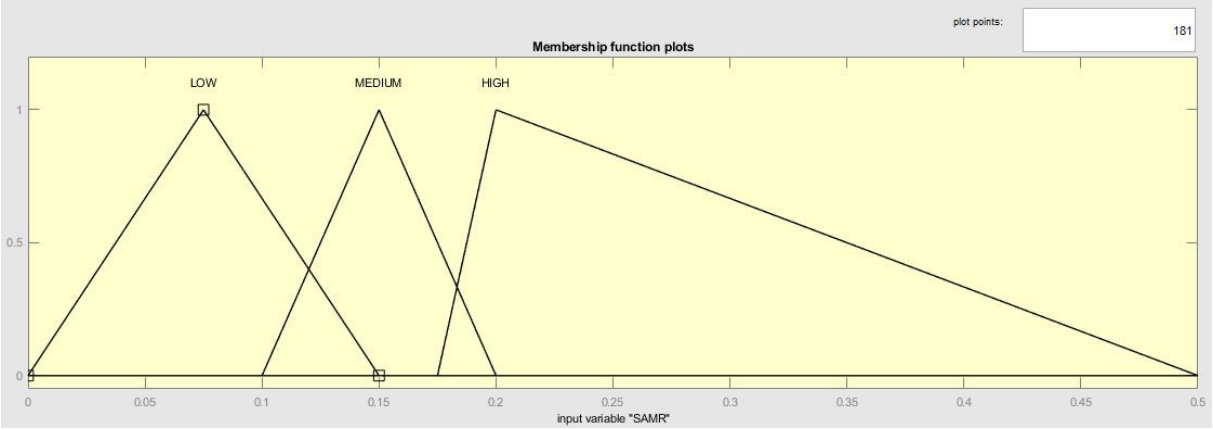

Figure 7. SAMR membership function

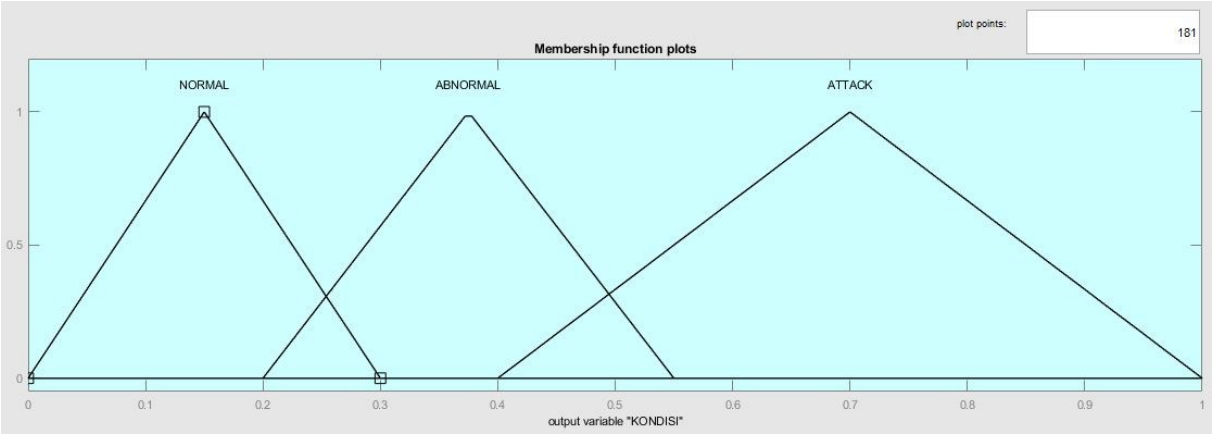

Figure 8. Output membership function

4.2. False Positive Ratio (FPR)

False Positive Ration (FPR) is used to determine the fuzzy error rate when detecting an attack when there is no attack on the network. The equation for calculating the FPR parameter is listed in Equation 5, where N_{FP} is the number of false positives and N_{TN} is the number of true negatives. Figure 9 shows the acquisition of the FPR value for each scenario. The lowest FPR is obtained in the 20-nodes scenario with a value of 0.1047; while the highest FPR is obtained in the 30-nodes scenario with a value of 0.4048.

$$FPR = \frac{N_{FP}}{(N_{FP} + N_{TN})} \quad (3)$$

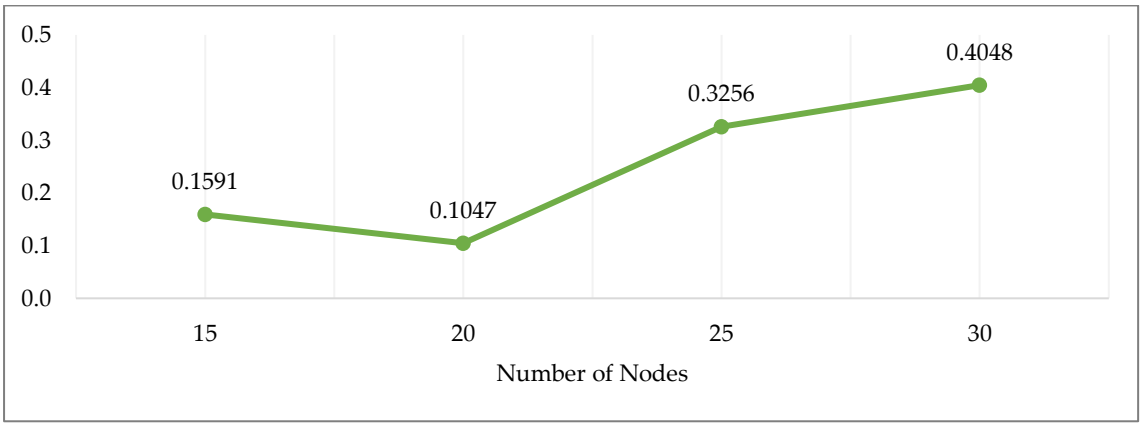

Figure 9. FPR value per scenario

ISSN 2502-3357 (Online)
---- et. al./Register xxx ISSN 2503-0477 (Print)

4.3. Accuracy

Accuracy is used to determine the success rate in detecting an attack fuzzy when an attack occurs or does not occur. The equation for calculating accuracy is shown in Equation 3, where N_{TP} is the number of true positives and N_{FN} is the number of false negatives. Figure 10 shows the accuracy for each scenario. The highest accuracy was obtained in the 20-nodes scenario during the attack interval of ATK 5S with a value of 94.74%. In comparison, the lowest accuracy is obtained in the 30-nodes scenario during the ATK 9S attack interval with a value of 79.29%.

$$\text{Accuracy} = \frac{N_{TP} + N_{TN}}{(N_{TP} + N_{TN} + N_{FP} + N_{FN})} \times 100\% \quad (4)$$

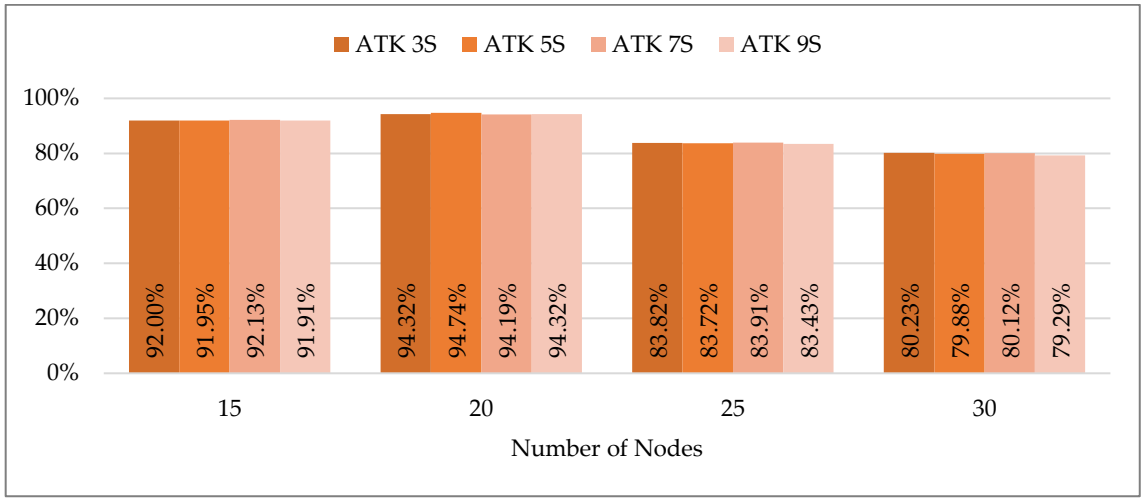

Figure 10. Accuracy percentage per scenario

4.4. Precision

Precision is used to determine the number of attack detection results identified correctly among all attack detection results. The equation for calculating precision parameters is listed in Equation 5. Figure 11 shows the acquisition of precision values based on the test results for each scenario. The highest precision was obtained in the 20-nodes scenario at the attack interval of ATK 3S and ATK 9S with a value of 0.9082. The lowest precision is obtained in the 30-nodes scenario at the ATK 9S attack interval with a value of 0.7119.

$$\text{Precision} = \frac{N_{TP}}{(N_{TP} + N_{FP})} \quad (5)$$

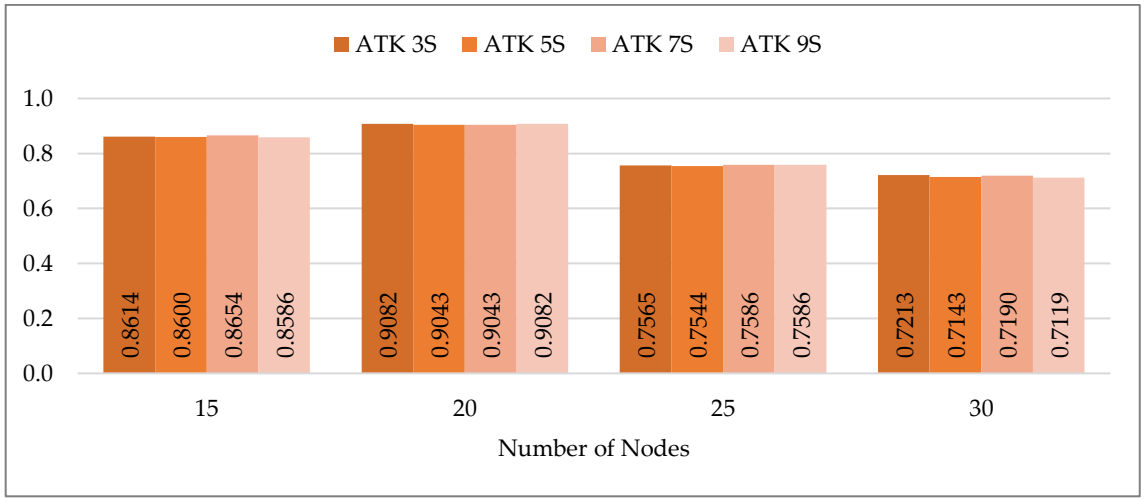

Figure 11. Precision value per scenario

ISSN 2502-3357 (Online)
MS Budiana et. al./Register xxx ISSN 2503-0477 (Print)

4.5. Recall

Recall is used to determine the number of attack detection results that are correctly identified at the attack time. The equation for calculating the recall parameter is as shown in Equation 6. Figure 12 shows the recall value obtained from the test results for each scenario. The recall value in each scenario has relatively the same value. Suppose the recall value of each scenario is averaged. In that case, the best average recall value is obtained in the 15-nodes scenario with an average value of 1. In comparison, the lowest average recall value is obtained in the 20-nodes scenario with an average value of 0.9914.

$$\text{Recall} = \frac{N_{TP}}{(N_{TP} + N_{FN})} \quad (6)$$

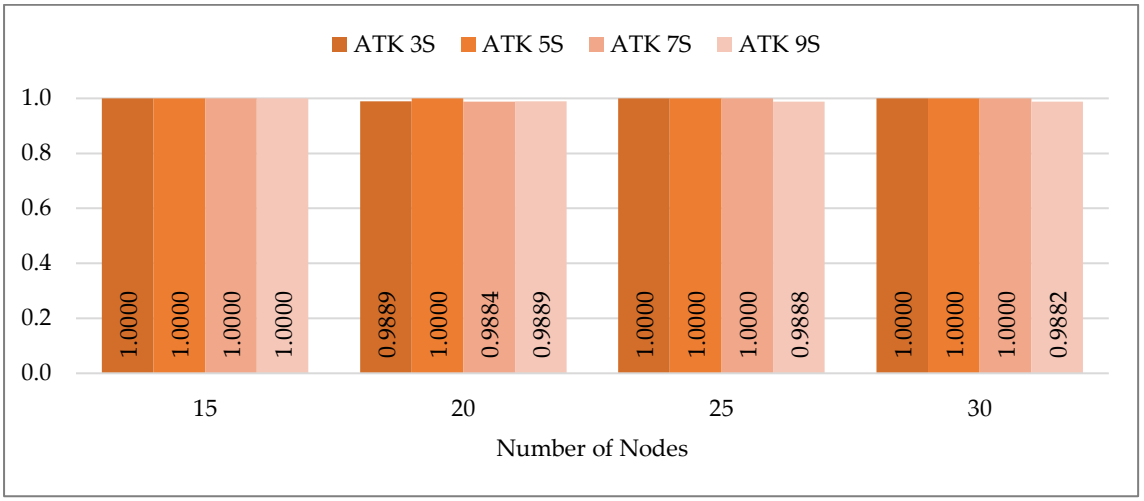

Figure 12. Average recall per scenario

4.6. F-Score

F-score used to average the precision with recall. The equation for calculating the f-score parameter is listed in Equation 7. Figure 13 shows the f-score value obtained from each test scenario. The highest F-score was obtained in the 20-nodes scenario during the attack interval of ATK 5S with a value of 0.9497; while the lowest f-score is obtained in the 30 nodes scenario at the ATK 9S attack interval with a value of 0.8276.

$$\text{F-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

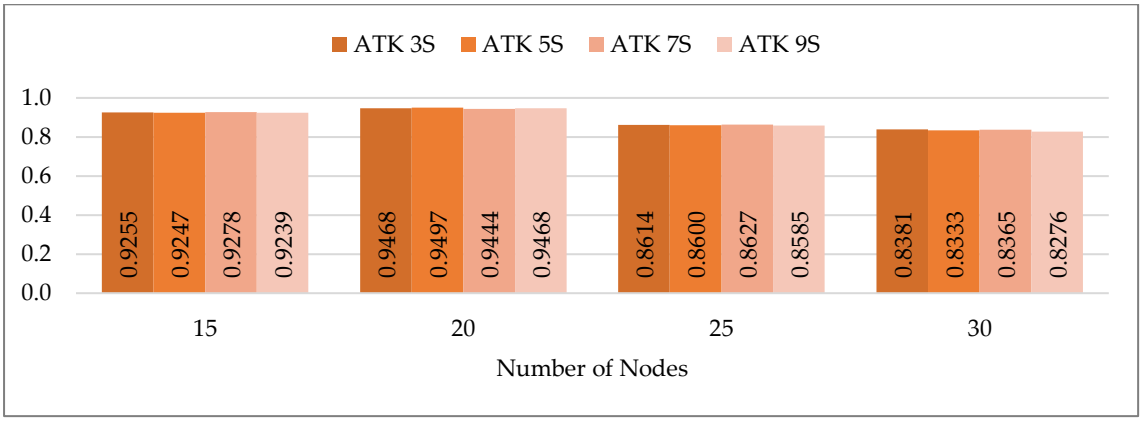

Figure 13. F-Score value per scenario

4.7. Fuzzy Logic Algorithm Analysis

Table 4 shows the average value of the performance of the fuzzy logic algorithm in each scenario. When there is no attack, the fuzzy logic algorithm's best performance is obtained in the 20-nodes scenario, which yields the lowest FPR. Meanwhile, when there is an attack, the fuzzy logic algorithm's

ISSN 2502-3357 (Online)

ISSN 2503-0477 (Print)

best performance is obtained in the 15-nodes scenario, giving the highest recall. The reason why the 20-nodes gives the best performance when there is no attack whereas the 15-nodes is better during an attack is due to the limitation of the fuzzy nodes, which can only accept a maximum of two topics simultaneously. There was very little traffic intensity in the 15-nodes scenario when there was no attack. This causes the fuzzy nodes to receive more than two topics at the same time (exceeding the receiving capacity), so the fuzzy nodes will discard that information from the broker. Meanwhile, the traffic intensity in the 15-nodes scenario when an attack occurs is still within the fuzzy nodes' receiving capacity, so that the fuzzy nodes can well receive the topics sent by the broker. In contrast to the 20-nodes scenario when there is no attack, the traffic intensity are still within the fuzzy nodes' receiving capacity so that all information sent by the broker can be well received. However, when an attack occurs in the 20-nodes scenario, the traffic intensity is high enough, causing delays in exchanging information between brokers and fuzzy nodes. This results in an error on the fuzzy nodes when filtering traffic and detecting DoS attacks. Therefore, the highest FPR is obtained in the 20-nodes scenario, and the highest recall is obtained in the 15-nodes scenario.

Table 4. The average performance of fuzzy logic per scenario

Scenarios	FPR	Accuracy	Precision	Recall	F-Score
15 Nodes	0.1591	92.00%	0.8613	1,0000	0.9255
20 Nodes	0.1047	94.39%	0.9062	0.9915	0.9469
25 Nodes	0.3256	83.72%	0.7570	0.9972	0.8607
30 Nodes	0.4048	79.88%	0.7166	0.9971	0.8339

According to table 4, it can be seen that the highest detection results for attacks that were correctly identified when an attack occurred or when an attack did not occur were obtained in the 20 nodes scenario. In comparison, the lowest level of attack detection results was obtained in the 30 nodes scenario. It can be seen from the acquisition of precision shown in Table 4. Overall, the fuzzy logic algorithm's best average performance when an attack occurs or does not occur is obtained in the 20 nodes scenario. In comparison, the worst fuzzy logic algorithm's average performance is obtained in the 30 nodes scenario. It can be seen from the f-score acquisition shown in table 2. This is because in the 20 nodes scenario, the fuzzy logic algorithm's performance when an attack does not occur. It can be seen from the low FPR in the 20 nodes scenario so that the accuracy, precision, and f-score parameters will be the highest. Meanwhile, in the 30 nodes scenario, the fuzzy logic algorithm's performance is terrible when there is no attack. It can be seen from the high FPR in the 30 nodes scenario so that the accuracy, precision, and f-score parameters will be the lowest. Meanwhile, in the 30 nodes scenario, the fuzzy logic algorithm's performance when there is no attack is very bad. It can be seen from the high FPR in the 30 nodes scenario so that the accuracy, precision, and f-score parameters will be the lowest. Meanwhile, in the 30 nodes scenario, the fuzzy logic algorithm's performance when there is no attack is very bad. It can be seen from the high FPR in the 30 nodes scenario so that the accuracy, precision, and f-score parameters will be the lowest.

5. Conclusions

The ability of fuzzy nodes to receive topics simultaneously amounts to a maximum of two topics. This can cause errors in the fuzzy logic algorithm when detecting DoS attacks. FPR is inversely proportional to accuracy, precision, and f-score. The highest FPR is obtained in the 30-nodes scenario with a value of 0.4048, while the lowest FPR is obtained in the 20-nodes scenario with a value of 0.1047. Accuracy, precision, and f-score in the 30-nodes scenario gave the lowest values, namely 79.88%, 0.7166, and 0.8339. In contrast, precision and f-score in the 20-nodes scenario yielded the highest values, namely 94.39%, 0.9062, and 0.9469. Furthermore, the highest average recall is obtained in the 15-nodes scenario

ISSN 2502-3357 (Online)
MS Budiana et. al./Register xxx ISSN 2503-0477 (Print)

with a value of 1, which means that all detection results of the fuzzy logic algorithm when an attack occurs in the 15-nodes scenario have a perfect detection rate. Future developments can use other fuzzy controller methods such as the Takagi-Sugeno method or the Tsukamoto method.

Author Contribution

M Soebagja developed model design. Ridha Muldina Negara validating research output and responsible for research activities. Arif Indra I formulated goal of research and assisting in developing model design. Harashta Tatimma Larasati reviewed and edited the total manuscript.

References

[1] A. P. Haripriya, "Secure-MQTT : an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things," 2019.
[2] A. Velinov and A. Mileva, "Running and Testing Applications for Contiki OS Using Cooja Simulator," no. June 2016, 2018.
[3] M. Yassine, A. Ezzati, and M. Belaissaoui, "An Enhanced DTLS Protocol for Internet of Things Applications," no. February 2018, 2016.
[4] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things," no. October, 2013.
[5] W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," *Futur. Gener. Comput. Syst.*, vol. 96, pp. 481–489, 2019.
[6] M. S. Harsha, B. M. Bhavani, and K. R. Kundhavai, "Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs," *2018 Int. Conf. Adv. Comput. Commun. Informatics*, pp. 2244–2250, 2018.
[7] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System," no. May, 2018.
[8] S. Shin, K. Kobara, C. C. Chuang, and W. Huang, "A security framework for MQTT," *2016 IEEE Conf. Commun. Netw. Secur. CNS 2016*, pp. 432–436, 2017.
[9] G. Potrino and A. F. Santamaria, "Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker," *2019 IEEE Wirel. Commun. Netw. Conf.*, pp. 1–6, 2019.
[10] S. Hameed, F. I. Khan, and B. Hameed, "Understanding Security Requirements and Challenges in Internet of Things (IoT): A Review," *Journal of Computer Networks and Communications*, vol. 2019. 2019.
[11] G. Potrino, F. De Rango, and P. Fazio, "A Distributed Mitigation Strategy against DoS attacks in Edge Computing," *Wirel. Telecommun. Symp.*, vol. 2019-April, pp. 1–7, 2019.
[12] A. P. Kelton, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. De, "Internet of Things : A survey on machine learning-based intrusion detection approaches," *Comput. Networks*, vol. 151, pp. 147–157, 2019.
[13] G. Karatas, "Deep Learning in Intrusion Detection Systems," no. February 2019, 2018.
[14] M. S. Munir, I. S. Bajwa, and S. M. Cheema, "An intelligent and secure smart watering system using fuzzy logic and blockchain ☆," *Comput. Electr. Eng.*, vol. 77, pp. 109–119, 2019.
[15] S. K. Yee and J. V Milanovi, "Fuzzy Logic Controller for Decentralized Stabilization of Multimachine Power Systems," vol. 16, no. 4, pp. 971–981, 2008.
[16] M. Alali, A. Almogren, M. M. Hassan, I. A. L. Rassan, and Z. Alam, "Improving Risk Assessment Model of Cyber Security Using Fuzzy Logic Inference System," *Comput. Secur.*, 2017.
[17] S. Dick, "Toward Complex Fuzzy Logic," vol. 13, no. 3, pp. 405–414, 2005.
[18] A. Machado *et al.*, "A Fuzzy Inference System to Support Medical Diagnosis in Real Time," *Procedia Comput. Sci.*, vol. 122, pp. 167–173, 2017.
[19] K. Ozera, K. Bylykbashi, Y. Liu, and L. Barolli, "A Fuzzy-Based Approach for Cluster Management in VANETs: Performance Evaluation for Two Fuzzy-Based Systems," *Internet of*

ISSN 2502-3357 (Online)

ISSN 2503-0477 (Print)

Things, 2018.

[20] A. F. Santamaria and F. De Rango, "A real IoT device deployment for E-Health applications under lightweight communication protocols, activity classifier and Edge data filtering," *Comput. Commun.*, 2018.

[21] S. H. Ramos, M. T. Villalba, and R. Lacuesta, "MQTT Security : A Novel Fuzzing Approach," vol. 2018, 2018.

[22] Vaccari, I., Aiello, M., & Cambiaso, E. (2020b). SlowTT: A slow denial of service against IoT networks. *Information (Switzerland)*, *11*(9), 1–18. https://doi.org/10.3390/INFO11090452

[23] SH Ramos, MT Villalba, and R. Lacuesta, "MQTT Security: A Novel Fuzzing Approach," vol. 2018, 2018.