7 (2) July 2021 95-106 ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
Register: *Jurnal Ilmiah Teknologi Sistem Informasi*

Contents lists available at www.journal.unipdu.ac.id

Register

Journal Page is available to www.journal.unipdu.ac.id/index.php/register

Research article

Advanced detection 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 [d]*

[a,b,c] *School of Electrical Engineering, Telkom University, Bandung, Indonesia*
[d] *School of Computer Science and Engineering, Pusan National University, Busan, Republic of Korea*
[d] *School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung 40116, Indonesia*
email: [a] mochsoebagja@gmail.com, [b,] ridhanegara@telkomuniversity.ac.id, [c] arifirawan@telkomuniversity.ac.id, [d] harashta@pusan.ac.kr*
** Correspondence*

ARTICLE INFO

Article history:
Received 3 March 2021
Revised 26 March 2021
Accepted 14 April 2021
Available online 19 April 2021

Keywords:
denial of service
fuzzy-logic
IoT
message queuing telemetry transport
MQTT

Please cite this article in IEEE style as:
M. S. Budiana, R. M. Negara, A. I. Irawan and H. T. Larasati, " Advanced detection Denial of Service attack in the Internet of Things network based on MQTT protocol using fuzzy logic," *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 7, no. 2, pp. 95-106, 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, the number of cybercrimes on IoT networks will increase. 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. The use of fuzzy logic algorithms with the Intrusion Detection System (IDS) scheme is suitable for detecting DoS because of its simple nature. This paper uses a fuzzy logic algorithm embedded in a node to detect DoS in 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 fuzzy performance.

1. Introduction

The Internet of Things (IoT) network or Machine-to-Machine (M2M) communication has become an essential part of today's era [1]. IoT can communicate 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]. Entities on an IoT network must have limited resources (such as memory, storage, or power) [3]. In general, IoT devices have an 8-bit microcontroller specification with 20kB RAM and 100kB ROM [2]. For the device to be able to connect to the internet network, addressing using IP is required. Therefore the researchers addressed IoT devices using 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].

The MQTT protocol has 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]. Publish/subscribe communication has three main elements: the publisher, subscriber, and broker. Publishers (for example, sensors, PCs, smartphones)

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

provide data and publish topics to brokers. Subscribers (for example, 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]. The light and simple characteristics of the MQTT cause problems on the safety side [10], so MQTT uses SSL/TLS as its standard security to prevent eavesdropping attacks or prevent a data damage. However SSL/TLS cannot handle DoS attacks [11]. One way to avoid DoS attacks is to use an Intrusion Detection System (IDS) [9]. IDS implementation can be done using the following algorithms: fuzzy logic [1], machine learning [12], deep learning [13], or blockchain [5]. Applying the fuzzy logic algorithm to the MQTT protocol is a suitable choice because it has several advantages. First, have a good approach to decision-making problems. Second, easy and simple implementation [14]

The fuzzy logic algorithm was invented by Lotfy Zadeh in 1965 [15, 16], which Lukasiewicz had studied for the first time in 1920 as many-valued logic [17]. Fuzzy logic is a development of boolean logic [18], 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 to fuzzy logic, which has degrees or degrees from 0 to equal to 1. Fuzzy logic will give output 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 output in the form of decisions to be made [19, 20].

This paper will discuss fuzzy logic algorithms' performance when detecting DoS attacks on IoT networks based on the MQTT protocol. The fuzzy logic algorithm will be planted on one of the network nodes as a subscriber and will detect DoS attacks with feature selection nodes.

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] fuzzy logic-based IDS was used to identify network anomalies against DoS attacks based on PUBLISH and SUBSCRIBE traffic from each client node. Both traffic will be used as input variables by fuzzy. Fuzzy will calculate the degrees of the two variables. It will execute the logic "IF ... THEN ..." using the fuzzy inference system (FIS); the FIS method used is the Mamdani method. Besides, this study uses fuzzy rule interpolation to generate new rules based on network traffic flow behavior. This research will produce a system capable of detecting DoS attacks on the MQTT protocol.

Harsha et al. In the paper[6] discuss the identification of security holes in MQTT using the 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, so the network is vulnerable to sniffing or data modification. The prevention 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. Meanwhile, prevention to increase data authority is to use 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 is used to keep the connection alive by avoiding connection closures by the server. The results of these attacks indicate that these attacks can exploit vulnerabilities in the MQTT network whether communications encrypted or not.

Andy et al. In the paper [7] discuss the handling of sniffing attacks, data modification, and Botnet on MQTT. Andy et al. 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), Andy et al. propose to put MQTT broker on a public network. Potrino et al. In the paper [9] proposes Host-based IDS (HIDS) attached to fog nodes to handle SYN flooding attacks, CONNECT flooding attacks, High

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

QoS message attacks, and DoS using PUBLISH attacks on MQTT. HIDS will identify and validate the buffer length and QoS level of CONNECT and PUBLISH messages coming from each client. Ramos et al. In the paper [23] propose a framework using template-based fuzzing techniques to detect DoS attacks on MQTT. This study aims to detect DoS attacks on the MQTT protocol using fuzzy logic algorithms. SUBSCRIBE, and SUBACK variables will be used as input variables in fuzzy logic. Fuzzy logic will be placed on network nodes and will monitor MQTT traffic flow with Feature Selection nodes.

3. Method

3.1. System design

The fuzzy logic algorithm will be embedded in one of the nodes, which is called a fuzzy node. Also, five additional nodes are needed as support nodes to observe the network's MQTT traffic flow as each node is only capable of subscribing to one topic. Fig. 1 shows a 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. The information needed by the feature selection nodes those are 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. So, that the fuzzy node can find out information about the MQTT traffic flow using topics that 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 Eq. 1, and the SAMR value using Eq. 2. The SMR defines the ratio between the number of SUBSCRIBE packets with all input packets that enter the broker, whereas the SAMR defines the ratio between the number of SUBACK packets with all output packets that leave the broker.

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

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

The SMR and SAMR values will be entered into a fuzzy logic algorithm and will be 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.

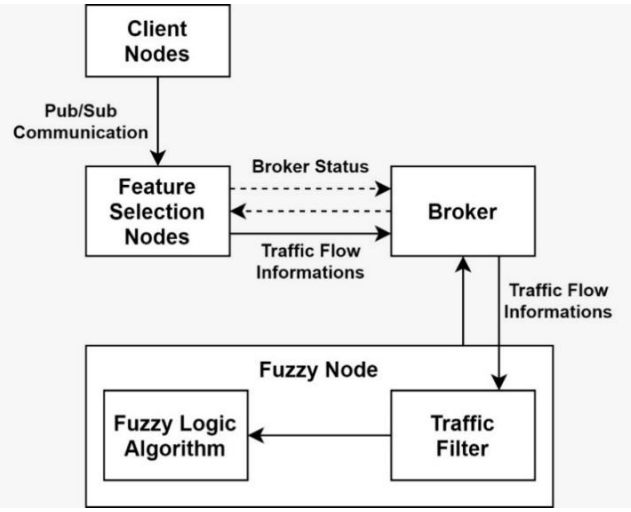

Fig. 1. System design

3.2. Feature Selection (FS) nodes

Nodes Feature Selection (FS) is responsible for observing the MQTT traffic flow on the network. Fig. 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

 http://doi.org/10.26594/register.v7i2.2340

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

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 superficial characteristics so that each node cannot subscribe to more than one topic.

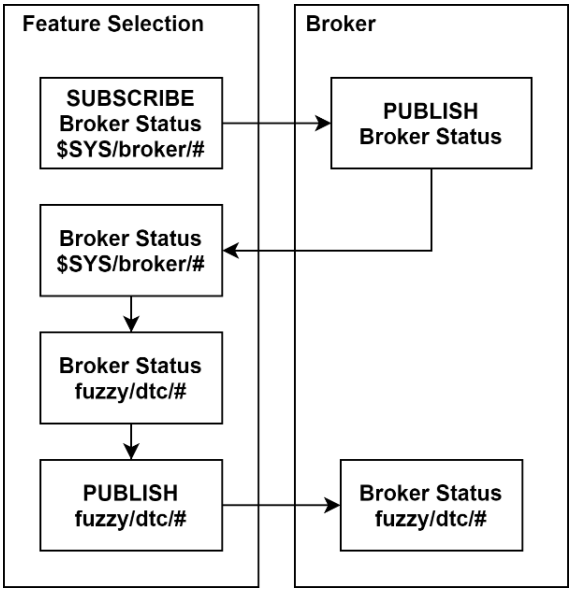

Fig. 2. FS model nodes

3.3. Fuzzy node

Fuzzy nodes are used to detect DoS attacks. Fig. 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 Eq. 1 and Eq. 2 to be used as input variables in the fuzzy logic algorithm. The fuzzy node will calculate the ratio between SUBSCRIBE packets with the number of input packets using Eq. 1 called SMR and the ratio between SUBACK packets with the number of output packets using Eq. 2 called SAMR.

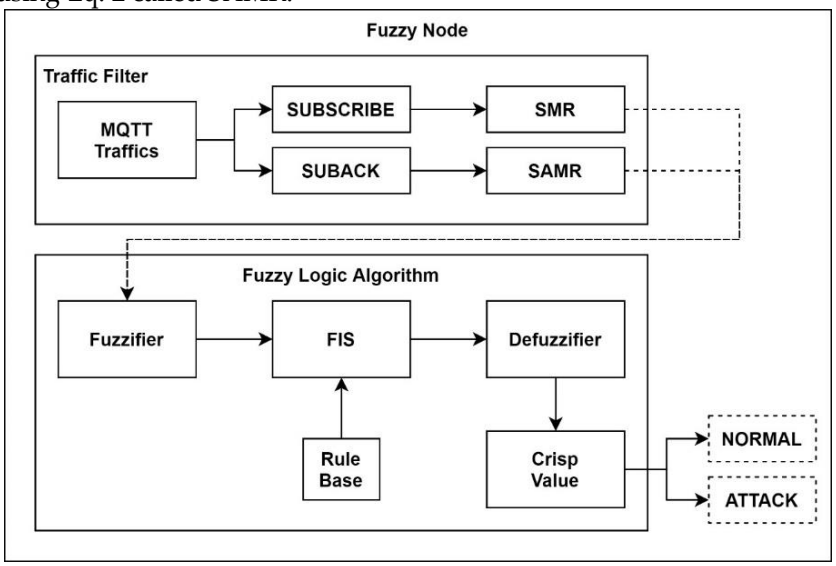

Fig. 3. Fuzzy node model

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

 http://doi.org/10.26594/register.v7i2.2340

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

"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, among others, "NORMAL" which represent the network is safe, "ABNORMAL" which represent there is suspicious activity on the network, and "ATTACK" which represent 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 was carried out using 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 is carried out using various scenarios, namely 15, 20, 25, and 30 nodes, with each scenario using 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 Discussion

4.1. Membership Function (MF) of fuzzy logic algorithm

Table 2 shows the SMR value obtained using Eq. 3. Table 3 shows the SAMR value obtained using Eq. 4. Based on Table 2 and Table 3, the SMR and SAMR values will increase along with the increase in the number of nodes or the increase in the attack's intensity. This attack is because the more the number of nodes, the traffic on the network 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 highest SAMR value was obtained during the ATK 3S attack interval at 15 nodes 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

Fig. 4 and Fig. 5 show a graph of the mean SMR and SAMR values. This condition is done so that MF SMR and SAMR can 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 Fig. 4 and Fig. 5, the forms of MF SMR and SAMR used to detect DoS attacks are shown in Fig. 6 and Fig. 7.

Determination of the form of MF output is obtained based on the experimental results using the

 http://doi.org/10.26594/register.v7i2.2340

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

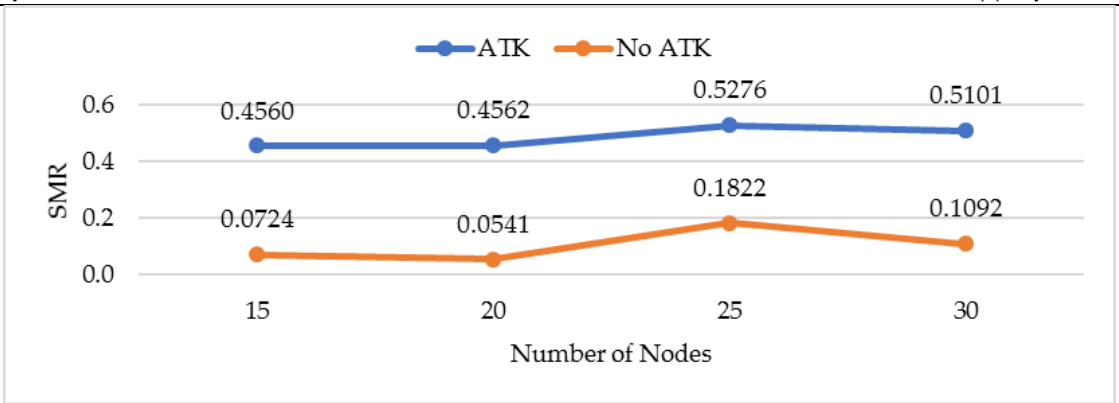

Fig. 1. Average of SMR value

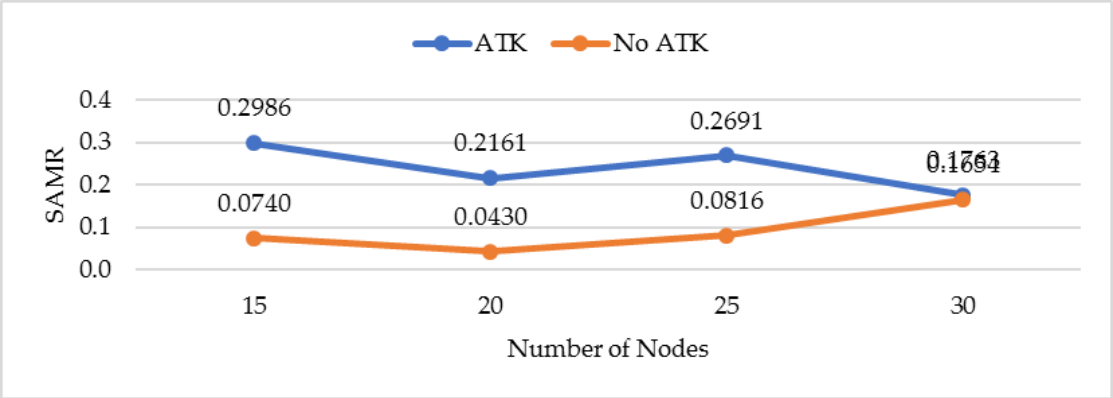

Fig. 2. Average of SAMR value

rule bases 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 network conditions, when in a state, 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. Fig. 8 is the MF of the output variable, which shows the network condition where if the crisp value is between zero and less than the reference value ($0 \leqslant$ crisps <0.375), it is assumed that there is no DoS attack on the network. If the crisp value is more significant than equal to the reference value until it is less than equal to one ($0.375 \leqslant$ crisps $\leqslant 1$), it is assumed that a DoS attack is taking place on the network.

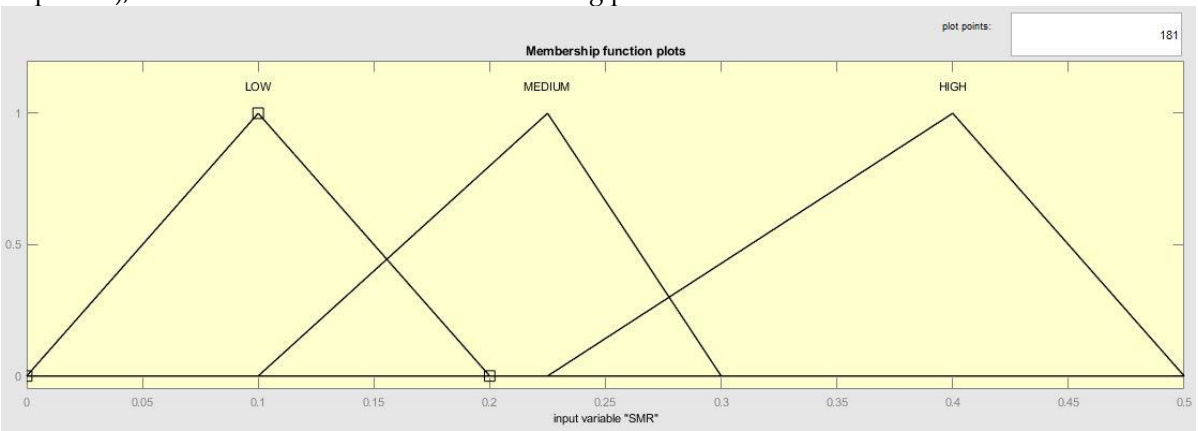

Fig. 3. SMR membership function

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 Eq. 5 where

http://doi.org/10.26594/register.v7i2.2340

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

NFP is the number of false positives and NTN is the number of true negatives. Fig. 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.

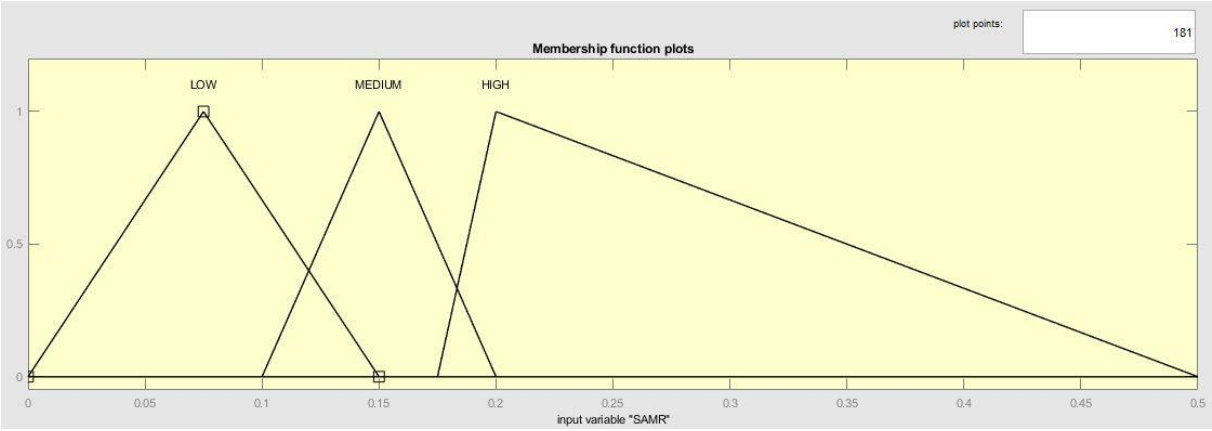

Fig. 4. SAMR membership function

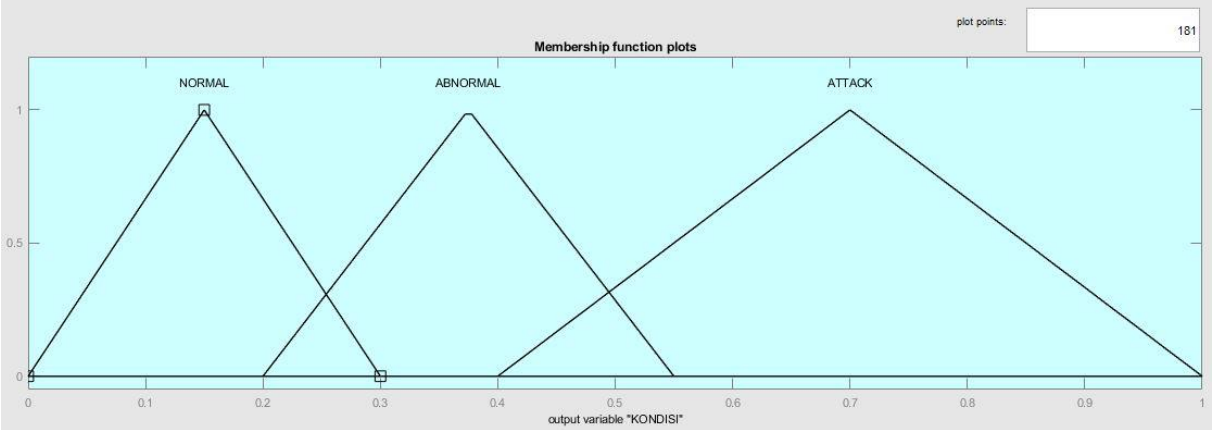

Fig. 5. Output membership function

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

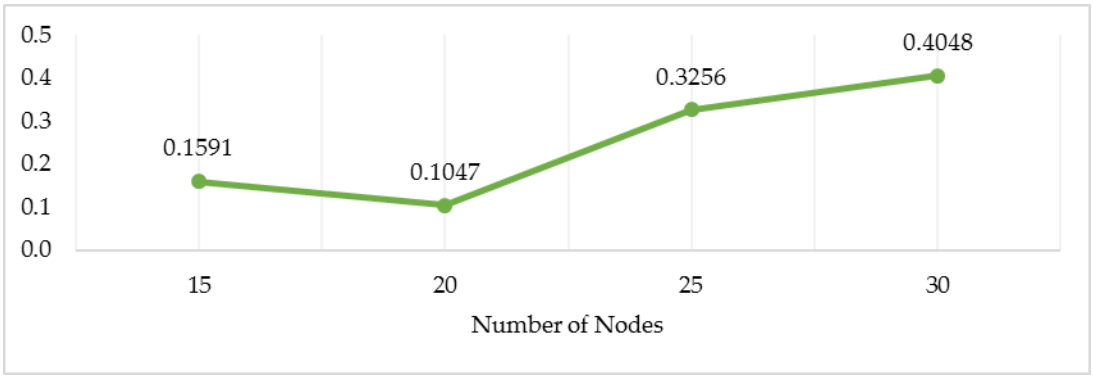

Fig. 6. FPR value per scenario

4.3. Accuracy

Accuracy used to determine the percentage of success in detecting an attack fuzzy when an attack occurs or does not occur. The equation for calculating accuracy is shown in Eq. 3 where NTP is the number of true positives and NFN is the number of false negatives. Fig. 10 shows the accuracy percentage for each scenario. The highest accuracy was obtained in the 20 nodes scenario during the attack interval of ATK 5S with a percentage of 94.74%. In comparison, the lowest accuracy is obtained in the 30 nodes scenario during the ATK 9S attack interval with a percentage of 79.29%.

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

 http://doi.org/10.26594/register.v7i2.2340

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

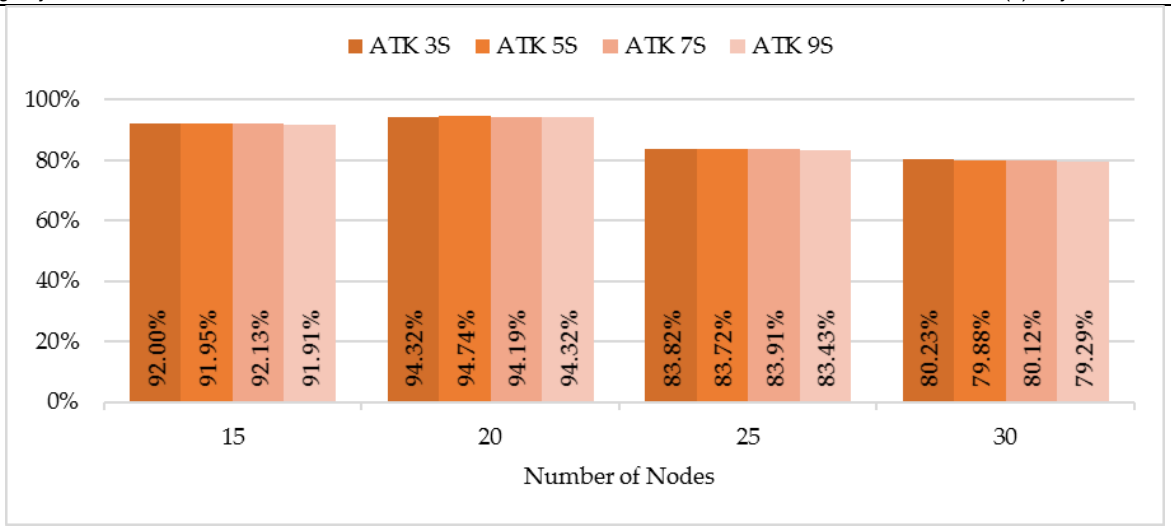

Fig. 7. Accuracy percentage per scenario

4.4. Precision

The 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 Eq. 5. Fig. 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.

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

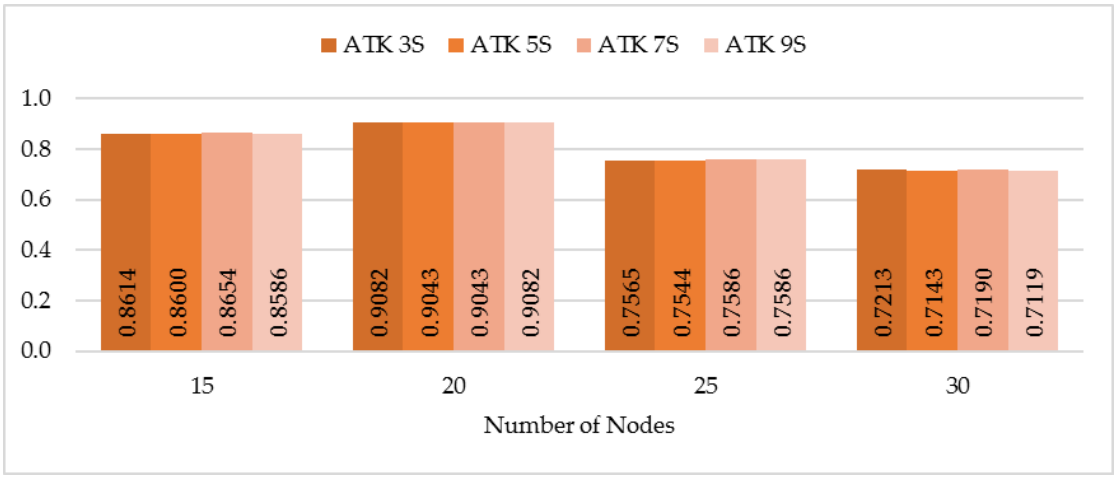

Fig. 8. Precision value per scenario

4.5. Recall

The 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 shown in Eq. 6. Fig. 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.

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

4.6. F-Score

F-score used to average the precision with recall. The equation for calculating the f-score parameter is listed in Eq. 7. Fig. 13 shows the f-score value obtained from each test scenario. The highest F-score was

 http://doi.org/10.26594/register.v7i2.2340

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

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.

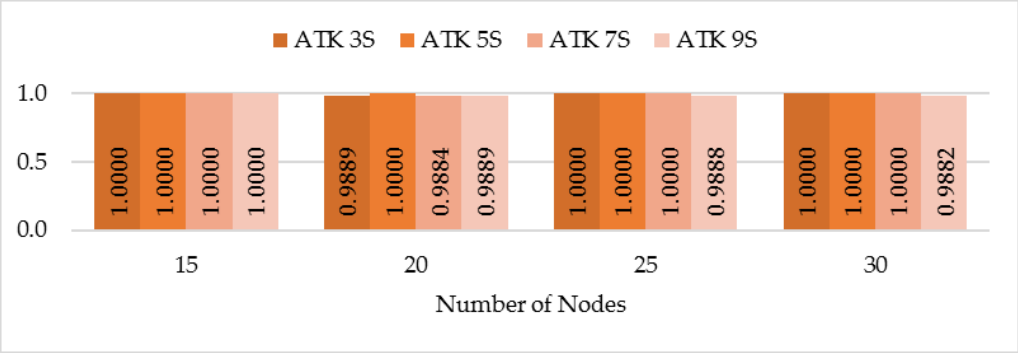

Fig. 9. Average recall per scenario

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

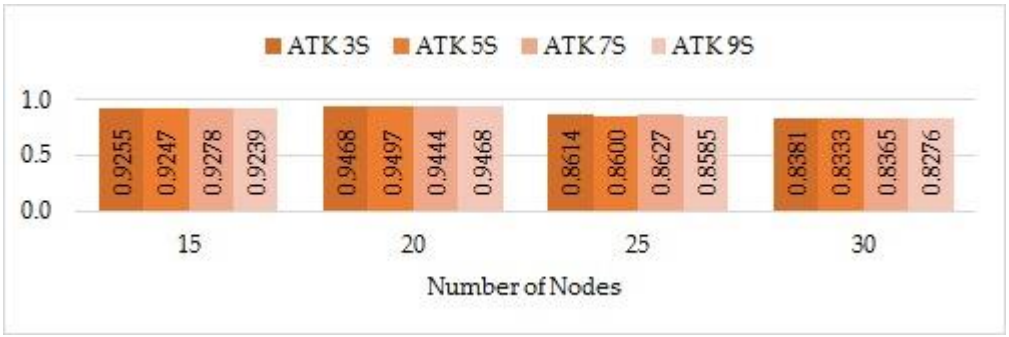

Fig. 10. F-Score value per scenario

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. It can be seen in Table 4 that the 20 nodes scenario has the lowest FPR. Meanwhile, when there is an attack, the fuzzy logic algorithm's best performance is obtained in the 15 nodes scenario. It can be seen in Table 4 that the 15 nodes scenario has the highest recall. This is due to the limitation of the fuzzy nodes that 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 follows 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 follows 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 creates 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.

Determination of the highest network condition that was correctly identified during the attack or when the attack did not occur was obtained in the 20 nodes scenario. In comparison, the determination of the lowest network condition was obtained in the 30 nodes scenario. It can be seen from the obtained accuracy shown in Table 4. 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.

http://doi.org/10.26594/register.v7i2.2340

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

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.

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

5. Conclusion

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 had the lowest values, namely 79.88%, 0.7166, and 0.8339. Accuracy, precision, and f-score in the 20 nodes scenario had the highest values, namely 94.39%, 0.9062, and 0.9469. The highest average recall is obtained in the 15 nodes scenario 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 Contributions

Mochamad Soebagja Budiana: Conceptualization, methodology, validation, and writing-original draft; Ridha Muldina Negara: Formal analysis and data curation; Arif Indra Irawan: Project Administration and funding acquisition; and Harashta Tatimma Larasati: Writing-original draft, writing-review & editing.

Declaration of Competing Interest

We declare that we have no conflict of interest.

References

[1] A. P. Haripriya and K. Kulothungan, "Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things," *J. Wireless Com. Network,* vol. 90, 2019.

[2] A. Velinov and A. Mileva, "Running and Testing Applications for Contiki OS Using Cooja Simulator," in *International Conference on Information Technology and Development of Education – ITRO 2016*, Zrenjanin, Republic of Serbia, 2016.

[3] Y. Maleh, A. Ezzati and M. Belaissaoui, "An Enhanced DTLS Protocol for Internet of Things Applications," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Fez, Morocco, 2016.

[4] P. Kasinathan, C. Pastrone, M. A. Spirito and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Lyon, France, 2013.

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

[5] W. Li, S. Tug, W. Meng and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," *Future Generation Computer Systems,* 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," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, India, 2018.

[7] S. Andy, B. Rahardjo and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Yogyakarta, Indonesia, 2017.

[8] S. Shin, K. Kobara, C.-C. Chuang and W. Huang, "A security framework for MQTT," in *2016 IEEE Conference on Communications and Network Security (CNS)*, Philadelphia, PA, USA, 2016.

[9] G. Potrino, F. d. Rango and A. F. Santamaria, "Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco, 2019.

[10] S. Hameed, F. I. Khan and B. H. R. J. o. C. N. a. C. 2019., "Understanding Security Requirements and Challenges in Internet of Things (IoT): A Review," *Journal of Computer Networks and Communications,* 2019.

[11] G. Potrino, F. D. Rango and P. Fazio, "A Distributed Mitigation Strategy against DoS attacks in Edge Computing," in *2019 Wireless Telecommunications Symposium (WTS)*, New York, NY, USA, 2019.

[12] K. A. d. Costa, J. P. Papa, C. O. Lisboa, R. Munoz and V. H. C. d. Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Computer Networks,* vol. 151, pp. 147-157, 2019.

[13] G. Karatas, O. Demir and O. K. Sahingoz, "Deep Learning in Intrusion Detection Systems," in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, Ankara, Turkey, 2018.

[14] M. S. Munir, I. S. Bajwa and S. M. Cheema, "An intelligent and secure smart watering system using fuzzy logic and blockchain," *Computers & Electrical Engineering,* vol. 77, pp. 109-119, 2019.

[15] S. K. Yee and J. V. Milanović, "Fuzzy logic controller for decentralized stabilization of multimachine power systems," *IEEE Transactions on Fuzzy Systems,* vol. 16, no. 4, pp. 971-981, 2008.

[16] M. Alali, A. Almogren, M. M. Hassan, I. A. Rassan and M. Z. A. Bhuiyan, "Improving risk assessment model of cyber security using fuzzy logic inference system," *Computers & Security,* vol. 74, pp. 323-339, 2018.

[17] S. Dick, "Toward complex fuzzy logic," *IEEE Transactions on Fuzzy Systems,* vol. 13, no. 3, pp. 405-414, 2005.

[18] I. B. d. Medeiros, M. A. S. Machado, W. J. Damasceno, A. M. Caldeira, R. C. d. Santos and J. B. d. S. Filho, "A Fuzzy Inference System to Support Medical Diagnosis in Real Time," *Procedia Computer Science,* 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 Things,* vol. 3–4, pp. 120-133, 2018.

[20] A. F. Santamaria, F. D. Rango, A. Serianni and P. Raimondo, "A real IoT device deployment for e-Health applications under lightweight communication protocols, activity classifier and edge data filtering," *Computer Communication,* vol. 128, pp. 60-73, 2018.

[21] I. Vaccari, M. Aiello and E. Cambiaso, "SlowITe, a Novel Denial of Service Attack Affecting MQTT," *Sensors,* vol. 20, no. 10, 2020.

[22] I. Vaccari, M. Aiello and E. Cambiaso, "SlowTT: A Slow Denial of Service Against IoT Networks," *Information,* vol. 11, no. 9, 2020.

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
7 (2) July 2021 95-106

[23] S. H. Ramos, M. T. Villalba and R. Lacuesta, "MQTT Security: A Novel Fuzzing Approach," *Wireless Communications and Mobile Computing*, 2018.