DoS attacks detection in MQTT networks

Dmitrii Dikii, Sergey Arustamov, Aleksey Grishentsev
Faculty of Secure Information Technologies, ITMO University, St Petersburg, Russia

Article Info

Article history:

Received Jun 5, 2020
Revised Aug 9, 2020
Accepted Aug 21, 2020

Keywords:

Artificial neural network
DoS attack
Internet of things
MQTT networks
Support vector machine

ABSTRACT

The paper considers the problem of protecting the Internet of things infrastructure against denial-of-service (DoS) attacks at the application level. The authors considered parameters that affect the network gateway workload: message frequency, payload size, number of recipients and some others. We proposed a modular structure of the attack detection tool presented by three classifiers that use the following attributes: username, device ID, and IP-address. The following types of classifiers have been the objects for the research: multilayer perceptron, random forest algorithm, and modifications of the support vector machine. Some scenarios for the behavior of network devices have been simulated. It was proved that for the proposed feature vector on simulated training and test data sets, the best results have been shown by a multilayer perceptron and a support vector machine with a radial basis function of the kernel and optimization with SMO algorithm. The authors also determined the conditions under which the selected classifiers have the best quality of recognizing abnormal and legitimate traffic in MQTT networks.

Corresponding Author:

Dmitrii Dikii
Faculty of Secure Information Technologies
ITMO University
St Petersburg, Russia
Email: dimandikiy@mail.ru

1. INTRODUCTION

Due to the rapid development of information technologies, information security issues are becoming particularly relevant. Currently, the one of the most rapidly progressing technology in IT industry is the Internet of Things (IoT). The basis of this technology is transmission of information between devices, and, consequently, associated threats that may be also topical for other network technologies. A distinctive feature of IoT is the usage of devices (sensors, actuators) that have quite limited resources such as energy consumption, for instance. This feature resulted in the development of energy-efficient protocols that can successfully transmit information over long distances (LoRa Protocol, as example). Application-level protocols tend to have fewer service information. One of the protocols that are specifically designed for IoT networks is the MQTT, the protocol being researched in this paper. This protocol is most effective when one needs to send the same information from a single device to a group of recipients simultaneously. To reach the target, the protocol implements the "publish-subscribe" pattern.

When analyzing information security threats existing in IoT networks, we need to consider all the variety of possible vulnerabilities and associated threats. The exhausting classification of threats for IoT networks, considering all levels of the OSI model, includes the following domains [1]: authentication and encryption [2, 3], access control, trust management, security policy management, software security, security of mobile devices roaming between different networks, hardware compatibility of devices [4], protection against DoS attacks, protection against repeated message attacks, protection against energy depletion attacks of autonomous devices and software errors [5, 6]. One of the challanging threats to IoT networks is a DoS attack that may be also a part of more extended botnet network attack. The implementing of a DoS attack using the possibilities of MQTT protocol can seriously damage the infrastructure of the IoT network. Consequently, users will not be able to send and receive messages. This is very essential in many critical areas such as medicine, life support, agriculture and industry.

The purpose of this study is to create a method for detecting abnormal device behavior that can lead to DoS directly at the gateway of the IoT network. The most common reason for increasing the load on the MQTT gateway is the increase of number processed messages of publish type (PT messages), according to the MQTT protocol specification [7]. Since the MQTT protocol implements the "publisher-subscriber" pattern, the amount of information circulating in the network strongly depends on the number of senders and recipients, as well as on the QoS value. Many research groups studied the MQTT protocol as a testing area for DoS attacks research. A comparative analysis of the methods used to simulate DoS attacks is shown in Table 1.

Table 1. The comparison of related works studies

Parameters	Number of devices	TLS	Modeling method	OSI Layer	Evaluated parameter	Reference
QoS 1,2	200 publishers, 120 subscribers	-	Computer simulation	Application layer	Processing time	[8]
TCPpackets	9 devices	NA	Real physical network	Network layer	NA	[9]
-	5 devices	NA	Computer simulation	Network layer	NA	[10]
QoS 0, 1, 2	1 device	-	Real physical network	Application layer	Processing time	[11]
NA	NA	NA	Real physical network	Application layer	Processing time	[12]
QoS 0, 1, 2	1 publisher, 100 subscribers	+	Computer simulation	Application layer	CPU usage	[13]
QoS 0, 1, 2	100 publishers	NA	Computer simulation	Application layer	Processing time	[14]
QoS 0, 1, 2	10 publishers	-	Computer simulation	Application layer	Processing time	[15]
4 MB payload	2000 publishers	NA	Computer simulation	Network layer, Application layer	CPU usage	[16]
10 KB payload	2000 publishers	NA	Computer simulation	Network layer, Application layer	Processing time; CPU usage	[17]

Unlike other studies, in our research DoS attack detection has the global effect of a larger number of factors contributing to an increase of the workload on the gateway. Based on the analysis of MQTT protocol and the results of the experiments, we propose a traffic feature vector that will be used to detect DoS attacks in real IoT networks with machine learning methods applied.

2. PROPOSED METHOD

The process of transmitting a PT message consists of at least two stages: sending a message from the sender to the gateway, and relaying the message as many times as the number of recipients as shown in Figure 1.

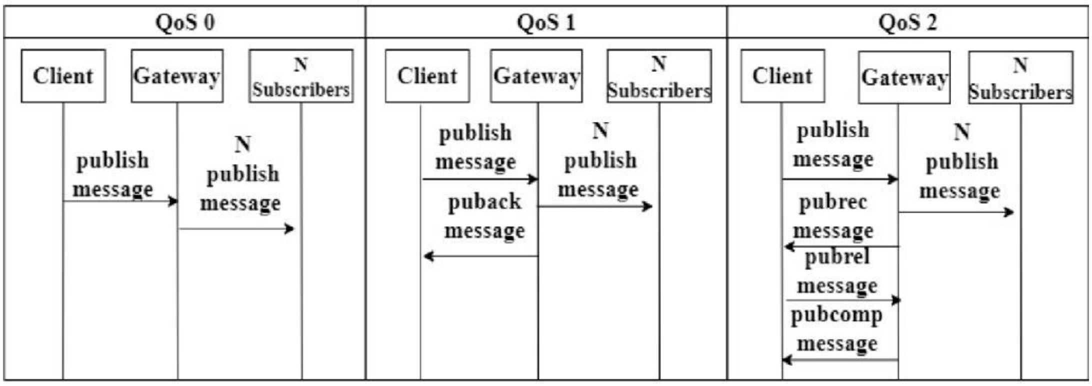


Figure 1. The process of transmitting information via publish message of the MQTT Protocol, where N is the number of subscribers

The main features of PT message include the following:
a) the title of the topic message;
b) the value of the QoS parameter;
c) the payload size;
d) the value of the dup flag (repeat message).

In addition to these parameters, we can specify the following service information for this type of message:
a) the network address of the sender;
b) the username of the sender;
c) the device ID of the sender;
d) the use of cryptographic transformations;
e) the number of subscribers.

Thus, a modular DoS attack detection system is proposed for detecting suspicious traffic over the MQTT protocol (Figure 2). This system collects and stores information related to a message, links traffic to network addresses, device ID, and username. In most cases, supervised algorithms based on machine learning are used as analyzer, so the system must have a learning module and store learning outcomes. To decrease the size of the data warehouse, one must clean it periodically removing obsolete information.

The operation of the modular attack detection system can be represented in the following way. At the first stage, the analyzers are trained on an existing database of legal and abnormal traffic, with all the necessary features for classification being extracted. The learning outcomes are saved to improve key operation indicators in the future (the learning process is indicated by dotted arrows in Figure 2). When the next PT message arrives at the system's input, its feature vector is extracted and stored in the database. Then the message data is sent to the input of three analyzers with pre-loaded settings obtained as an outcome of training. At the output of the analyzers, a decision is made on the traffic legitimacy. If an anomaly is detected, the source can be identified: a compromised network node, a compromised user account, or a compromised device. If a message has passed all checkouts, it is sent to the gateway and further to subscribers.

To design an attack detection system, the future vector that will be evaluated by the analyzers of a message, should be defined. Although increasing the QoS value is insignificant, it still affects the performance of the gateway. In addition, there is an obvious relationship between the number of subscribers and the speed of processing messages at the gateway, which was not revealed in the papers of other research groups.

Another important parameter of this message type is the size of the payload. As a rule, the data is presented either in plain form or in JSON format. XML format is used less frequently. The MQTT protocol supports sending messages up to 256 megabytes in size. To determine the impact of the payload size on the overall performance of the gate, an experimental simulation have been performed in the study. Messages with a fewer payload, around 80 bytes, were sent to the gate sequentially, and then messages with a high payload, about 80 Kilobytes, were sent in parallel from other devices. We measured the number of messages processed by the gate per second. In this case, the QoS value was equal to zero, with the number of subscribers being the same and equal to one. After analysis of the test results as shown in Figure 3 we observed a significant impact of the message payload size on the gateway load. Starting from the fifth second of testing, we received a sharp, approximately 4-5 times, dropdown in the number of messages processed per second. After we stop sending large messages, the gate performance is quickly recovered.

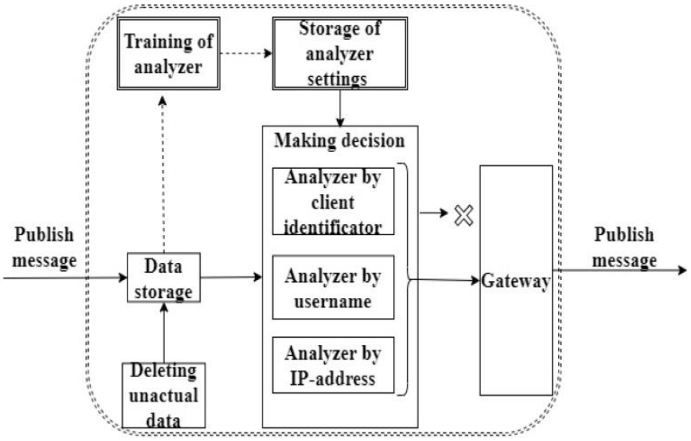


Figure 2. A block diagram of the detection system denial of service via the MQTT protocol

Thus, in traffic analysis, apart from the number of messages, it is very essential to consider their size, as well as the number of recipients. However, using standard network monitoring tools, there is no way to get information about the number of recipients without implementing changes to the gateway source code. The use of cryptographic transformations for message payload may also drastically impact the traffic analysis and operation indicators of the network.

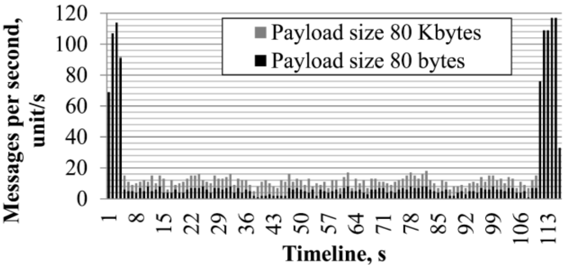


Figure 3. The comparison of gateway performance when processing messages with small and large payload sizes at the same time

Traffic analysis is usually performed using statistical methods and machine learning methods. In this paper, we consider such algorithms as an artificial neural network (multilayer perceptron), the support vector machine, and the random forest algorithm. These three types of algorithms are used to classify an object by a finite set of attributes. The main task of our research is to reveal the method that shows the best results of detecting abnormal behavior of devices based on the characteristics of the MQTT protocol traffic.

2.1. Artificial neural network

This model consists of a network of neurons that are divided into layers [18]. Layers are divided into input, hidden, and output layers. The number of input features of an object determines the number of neurons in the first layer. The output layer has a dimension equal to the number of classes. Neurons of neighboring layers have connections with each other, which are called weight coefficients. The value of a neuron depends on all incoming weights and values of the previous layer's neurons and is defined as:

$$y = f(\sum_{i=1}^{N} x_i w_i) \quad (1)$$

where xi – value of i-th neuron of the previous layer, wi is the value weight connecting i-th neuron of the previous layer. The activation function of the neuron y=f (x) can be represented as a linear, threshold, or sigmoidal function. This study used a model of a multilayer perceptron with the function of activating neurons in the form of a sigmoid:

$$f(x) = \frac{1}{1+e^{-x}} \quad (2)$$

The multilayer perceptron has to be trained to find the most appropriate values of weight coefficients. Training can be carried out using the back propagation algorithm or genetic algorithms.

2.2. The algorithm of random forest

The random forest algorithm is based on the approach implemented in the decision tree algorithm. The composition of the random forest consists of many trees [19]. Each decision tree in the forest is based on a sample obtained by the bootstrap method [20]. It is important to underline that the decision tree uses the same number of random features of the training sample to create a random forest consisting of different decision trees. The decision tree is based on CART, ID3, and C4.5 algorithms [21]. Training is performed in order to get two the most homogeneous subsamples for each root. Determining the homogeneity of subsamples may be performed in one of the following ways:

the calculation of the entropy:

$$I = -\sum_{i=1}^{v} P(w_j) log_2 P(w_j) \quad (3)$$

the calculation of the Gini index:

$$I = 1 - \sum_{i=1}^{v} P^2(w_j) \tag{4}$$

the calculation of the frequency of erroneous classification:

$$I = 1 - \max P(w_j) \tag{5}$$

where P(wj) is the probability of finding an object of class wj in the sample.

In our research we apply the entropy approach as the most straight through way of calculation (3).

2.3. The support vector machine

The support vector machine is often used to classify objects with a large-dimensional feature vector. This method is based on constructing a hyperplane in a multidimensional space in such a way that it separates objects of different classes as best as possible [22]. In the simplest case, the equation of the desired hyperplane takes the form:

$$(\bar{w}\bar{x}) - c = 0 \tag{6}$$

where $\bar{w}$ are coefficients of hyperplane, c is a constant.

If the value is less than zero (yi<0) for xi, then the object belongs to one class, otherwise to another. The task of training and optimization in the support vector machine is to construct a hyperplane that is equidistant from instances of all classes. The optimal hyperplane equation is be based on Langrangian function:

$$\sum_{s=1}^{k} a_{is}^{0} y_{is} (\overline{x_{\iota s}}\bar{x}) + b = 0 \tag{7}$$

where s=1...k are the support vectors, and ai are the Lagrangian multipliers.

The support vectors are the nearest sample points, which lie on the hyperplanes $(\bar{w}\bar{x}) + b = \pm 1$ if a0≥0.

The support vector machine uses the kernel function. It may be linear or nonlinear (the kernel function $K = (\bar{x_\iota}\bar{x})$ is substituted in the expression $((\bar{x_\iota}\bar{x})$ in equation (7)), that significantly affects the shape of the hyperplane. In this study radial based kernel function (RBF) and linear kernel function were considered. We need also separately distinguish the algorithm for optimizing the support vector machine – SVM [23].

One of the main indicators of the classifier that will allow us to evaluate its efficiency along with accuracy (8) is the F1-score [24]. This indicator depends on the classification results of the test dataset: TP- number of legal traffic messages correctly classified, TN- number of abnormal traffic messages correctly classified, FN- number of legal traffic classified as abnormal ones, and FP – number of abnormal traffic messages classified as legal ones.

$$\mathrm{A} = \frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

To calculate F1-score, one needs to calculate the precision (9) and recall (10):

$$P = \frac{TP}{TP+FP} \tag{9}$$

$$R = \frac{TP}{TP+FN} \tag{10}$$

The required value F1-score is defined by formula (11):

$$F = \frac{2*P*R}{P+R} \tag{11}$$

3. RESEARCH METHOD

Training and test traffic data sets for the MQTT protocol have been arranged with the software based on the open source project Moquette [25], for the hardware platform we have chosen Raspberry PI 3 model B microcomputer. The software code of the gateway was modified in JAVA according to the diagram shown in Figure 2. Two personal computers (PC1 and PC2) have been used as MQTT clients. The PC2 was responsible for receiving messages and the PC1 was sending them as shown in Figure 4. The one hundred clients on PC2 have been launched simultaneously using the "Paho-MQTT" library supported by Python programming language [26]. Before starting the experiment, each client of the PC2 was assigned an ID according to the sequence number. The sequence number corresponds to the number of topics the client is subscribed to.

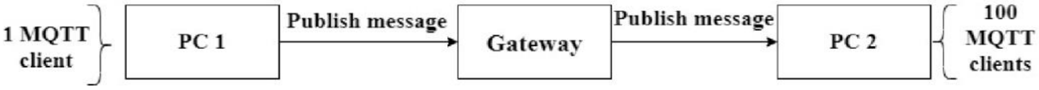


Figure 4. Diagram of an experimental setup for generating a training and test data set

The PC1 generated legal and abnormal traffic of PT messages with specified parameters of message frequency and other parameters. Once a message is received by the gateway, a corresponding feature vector is extracted. We considered the following traffic parameters: the delay between messages, the payload size, and the number of subscribers to the topic. The full feature vector is shown in Table 2. All message parameters can be retrieved when the gateway software is modified.

Table 2. Feature vector of message attributes using the MQTT Protocol

Definition	Parameter	Type
Username	User`s name	String
Client_ID	Device identifier	String
IP_address	Network address of device	String
TLS_enable	Usage of cryptographic transformations	Boolean
QoS	Quality of service	0, 1 or 2
Time	Time when a message was registered at the gateway	Ms
Payload_size	Payload size of messagr	Byte
Subscribers_N	Number of subscribers	Units

Thus, having one feature vector, all three classifiers can operate in parallel, as shown in Figure 2, using the appropriate label (Username, Client_ID, or IP_address). In order to reduce the number of fault positive and fault negative errors, an average feature vector of messages received over a time interval was submitted to the classifiers. To define the most appropriate time interval was one of the main problems in this research. We selected the following set of time intervals varying in the range (20; 20000) MS: 20, 50, 100, 250, 500, 1000, 1500, 2000, 5000, 10000, 15000, 20000.

3.1. Modeling test and training data set

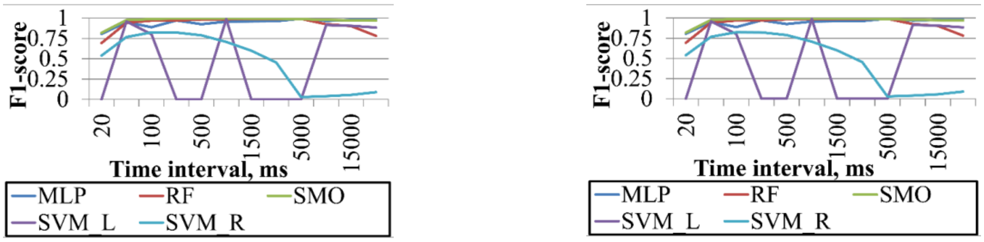
We simulated network traffic basing on the following parameters: message frequency, payload size, number of subscribers, QoS value, and use of cryptographic transformations. Two scenarios have considered for legal traffic. In the first one, the delay interval between messages belonged to a range of (0; 500) MS, the size of the payload (1; 80) Bytes, and the number of subscribers (1; 5) units. The second scenario was introduced with a long delay time between messages (0; 5000) MS, with the payload size (0; 800) Bytes, and the number of subscribers in the range of (1; 10).

Abnormal traffic was presented with a minimum delay between messages (0 MS) and /or a large payload size (60; 80) Kbytes and/or many subscribers (75; 100). Thus, traffic was recognized as being abnormal if at least one parameter corresponded to the above characteristics. For each scenario of legitimate traffic a corresponding abnormal traffic has been simulated. The test data set has been simulated in a similar way but time intervals for parameters fluctuation for legal and abnormal traffic have been extended.

4. RESULTS AND ANALYSIS

In order to determine the most suitable classification method on the same test and training data, we calculated F1-score values for traffic evaluation at different time intervals. Upon completion of the experiment, we obtained the following results. For the first scenario of legal traffic as shown in Figure 5(a) in the area from 50 MS to 1000 MS, we managed to achieve a value of F1-score greater than 0.8 for all classifiers. Compared classifiers can be split into two groups. The first group is the random forest algorithm and the support vector machine with RBF, where the value of the F1-score begins to decrease rapidly when the time interval increases by more than 1000 MS. The second group, consisting of the remaining classifiers, had F1-score value close to 0.9 with the same interval values. For small intervals (not exceeding 100 MS), the best results have been observed for the support vector machine with a linear kernel function, the support vector machine with the RBF and SMO optimization function, and the random forest algorithm. The multilayer perceptron showed good results at intervals longer than 500 MS. Thus, the graph of changes for F1-score of time intervals during which traffic is evaluated can be divided into three parts. The first one is a gradual increase in the value of the F1-score. The second part keeps the value of the F1-score at the same level with small deviations. The third part is a sharp dropdown of F1-score value. Moreover, the lowest decrease is observed in the method of support vector machine with RBF and SMO optimization and a multilayer perceptron.

The results of a similar experiment for the second legal traffic scenario are shown in Figure 5(b). The most significant difference from the first scenario is the change of F1-score for the support vector machine with a linear kernel function. Here, low values of this parameter are linked to a big number of false negative decisions of the classifier. As in the first scenario, one of the worst results was shown by support vector machine with RBF. For the random forest algorithm, the F1-score behavior is similar. With larger time interval the value of the F1-score begins to decrease. It is worth noting that the quality of classification tends to deteriorate when the time interval is more than three times of message delay limit for legal traffic for both scenarios. The support vector machine with RBF and the SMO optimization method, as well as the multilayer perceptron, showed the best results. With interval longer than 50 MS, the value of the F1-score for both classifiers have not fallen below the level of 0.90.



(a)

(b)

Figure 5. The dependence of the F1-score values against changes in the time interval during which the estimated traffic for: the first scenario (a); the second scenario (b); MLP – multilayer perceptron, RF - algorithm random forest, SMO – support vector machine with RBF and the SMO, SVM_L – support vector machine with a linear kernel function, SVM_R – support vector machine with RBF.

Thus, to detect a denial of service in the MQTT protocol networks with proposed feature vector the best solution is to apply support vector machine with RBF and optimization SMO method or artificial neural network in the form of multilayer perceptron. These classifiers showed approximately the same results, and the value of the F1-score decreases more slowly than in other algorithms considered when the traffic estimation interval increases. Other classifiers in some parts of the charts also showed good results, but, as the study revealed, require more careful calibration of the time interval during which traffic is being evaluated. In comparising with simular works, our approach demonstrates the better results in DoS attack detection for some indicators (optimal time interval and classification method). For instance, in our research F1-score value reached the value of 0.98 whereas the results of [27] shows only 0.90 and research results of [28] being equal to 0.95.

5. CONCLUSION

In this paper, we considered the problem of detecting Dos attacks in networks using the MQTT protocol. Simulation results showed that the workload of the gateway is impacted by many factors, some of which were overlooked in the earlier works of the research groups. The main parameters that an adversary can use to overload the gateway are message frequency, payload size, and number of recipients. To solve the problem of detecting a DoS attack in MQTT networks, we proposed a method based on machine learning algorithms. The proposed method showed best quality of attack detection and proved the effectiveness of our approach. The F1-score value of attack detection is higher or at the same level as others research group results. The experiment showed that for efficient traffic analysis, it is necessary to process traffic during the time longer than 250 MS. At this interval, the multilayer perceptron and the support vector machine with RBF and optimization based on SMO algorithm showed the best results. It is worth noting that an excessive increase in the interval during which traffic is analyzed leads not only to an increase in computational and time costs, but also negatively affects the quality of classification. Our research confirms that machine learning methods can be used as a tool for detecting cyber security attacks in MQTT networks. The results of this study can be applied in design of intrusion detection systems for IoT environments and further research into the security of publisher-subscriber networks.

ACKNOWLEDGEMENTS

The reported study was funded by Russian Ministry of Science (information security) №11/2020.

REFERENCES

[1] J. Granjal, et al., “Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues”, *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294-1312, 2015. https://doi.org/10.1109/comst.2015.2388550.

[2] S. Sicari, et al., “Security, privacy and trust in Internet of Things: The road ahead”, *Computer Networks*, vol. 76, pp. 146-164, 2015. https://doi.org/10.1016/j.comnet.2014.11.008.

[3] M. Daud, et al., “Denial of service: (DoS) Impact on sensors”, 2*018 4th International Conference on Information Management (ICIM)*, 2018, https://doi.org/10.1109/infoman.2018.8392848.

[4] E.N. Velichko,.et al., “Improvement of finite difference method convergence for increasing the efficiency of modeling in communications”, *Lecture Notes in Computer Science*, vol. 8638, pp. 591-597, 2014. https://doi.org/10.1007/978-3-319-10353-2_54.

[5] M.A. Khan and K. Salah, “IoT security: Review, blockchain solutions, and open challenges”, *Future Generation Computer Systems*, vol. 82, pp. 395-411, 2018, https://doi.org/10.1016/j.future.2017.11.022.

[6] M. Frey, et al., “Security for the Industrial IoT: The Case for Information-Centric Networking”, *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, https://doi.org/10.1109/wf-iot.2019.8767183.

[7] Official site MQTT. URL: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html. [Online; accessed 29.01.2020].

[8] B. Chifor, Patriciu V., “Mitigating DoS attacks in publish-subscribe IoT networks”, *2017 In Proceedings of Conference: Electronics, Computers and Artificial Intelligence. International Conference (ECAI 2017)*, 2017. https://doi.org/10.1109/ECAI.2017.8166463.

[9] Y. Meidan, et al.,”N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders”, *IEEE Pervasive computing*, vol. 13, no. 9, pp. 1-8, 2018. https://doi.org/10.1109/MPRV.2018.03367731.

[10] N. Koroniotis, et al., “Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset”, *Future Generation Computer Systems 100*, pp. 779-796, 2019. https://doi.org/10.1016/j.future.2019.05.041.

[11] V-D. Pham, et al., “Research of Protocols of Interaction of the Internet of Things on the Basis of the Laboratory Bench”, *Telecom IT*, vol. 4, no. 1 pp. 55-67, 2016 (in Russian).

[12] R. Dolgushev, et al., “An Overview of Possible Testing Types and Methods for the Internet of Things”, *Telecom IT*. vol. 4, no. 2 pp. 1-11, 2016 (in Russian).

[13] P. Fehrenbach, “Messaging Queues in the IoT under pressure”, *Computational Science and Its Applications – ICCSA*. 2018. pp. 1-9. URL: https://blog.it-securityguard.com/wp-content/uploads/2017/10/IOT_Mosquitto_Pfehrenbach.pdf [Online; accessed 18.06.2020].

[14] B. Mishra, “Performance Evaluation of MQTT Broker Servers”, *Lecture Notes in Computer Science*, vol 10963, pp. 599-609, 2018. https://doi.org/10.1007/978-3-319-95171-3_47.

[15] M. Handosa, D. Gracanin, “Performance evaluation of mqtt-based internet of things systems”, *In Proceedings of the 2017 Winter Simulation Conference*, pp. 4544-4545, 2017. https://doi.org/10.1109/WSC.2017.8248196.

[16] S.N. Firdous, et al., “Modelling and Evaluation of Malicious Attacks against the IoT MQTT Protocol”, *In Proceedings 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 748-755, 2017. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.115.

[17] C. Bao, et al., “A Tool for Denial of Service Attack Testing in IoT”, *In Proceedings 8th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 1-6, 2016.

[18] F. Rosenblatt, “The perceptron-a perceiving and recognizing automaton,” *Technical Report 85-460-1 Cornell Aeronautical Laboratory*, 1957.

[19] S.P. Chistiakov, “Random forests: an overview”, *Transactions of the Karelian Research Centre of the Russian Academy of Sciences*, vol. 1, pp. 117-136, 2013 (in Russian).

[20] B. Efron, “Bootstrap Methods: Another Look at the Jackknife,” *Annals of Statistics*, vol. 7, no. 1, pp. 1-26, 1979. https://doi.org/10.1214/aos/1176344552.

[21] I. D. Mienye,et al.,”Prediction performance of improved decision tree-based algorithms: a review,” *Procedia Manufacturing*, vol. 35, pp. 698-703, 2019, https://doi.org/10.1016/j.promfg.2019.06.011.

[22] V.V. V'jugin, “Mathematical foundations of the theory of machine learning and forecasting,” pp. 387, 2013. (in Russian).

[23] J. Platt, “Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines”, 1998.

[24] S.V. Voloshin, et al., “Analysis of the quality of binary classification web pages using the support vector method,” *News of Altai State University*, vol. 96, no. 4, pp. 84-88, 2017. (in Russian).

[25] Official site Moquette. URL: https://projects.eclipse.org/projects/iot.moquette. [Online; accessed 29.01.2020].

[26] Official site Paho-MQTT project. URL: https://pypi.org/project/paho-mqtt/. [Online; accessed 29.01.2020].

[27] A. P. Haripriya, Kulothungan K, “Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things”. *EURASIP Journal on Wireless Communications and Networking*, pp. 1-15, 2019. https://doi.org/10.1186/s13638-019-1402-8.

[28] H. Alaiz-Moreton, et al., “Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol”, *Complexity*, pp. 1-11, 2019. https://doi.org/doi:10.1155/2019/6516253.