An Efficient Intrusion Detection System for Attacks Detection in MQTT Protocol Using Machine Learning

Tahani Gazdar

Cybersecurity Department, College of Computer Science and Engineering
University of Jeddah, Jeddah, Saudi Arabia

Summary

IoT has gained a lot of attention during the last decade. Although the applications provided in an IoT environment are supposed to simplify and automate our everyday tasks, their deployment is facing many security challenges. The high number of connected smart devices which are enabled with sensing and communication capabilities, in addition to the inherent connectivity to the Internet and the high amount of collected data might cause different attacks to emanate. Intrusion detection systems are one of the important defense techniques against attacks. In IoT, lightweight, accurate, and efficient IDSs are required. Many studies show that ML has the potential to accurately detect attacks given adequate data. Although distinct efforts have been made in designing intrusion detection systems for IoT, IDS based on Machine learning models are still in the early stages. More important, few IDS consider detecting attacks specific to IoT protocols like MQTT, CoAP, DDS, AMQP, etc. Almost all studies suppose that IoT applications are based only on HTTP traffic. Apart from that, almost all intrusion detection models designed for IoT are not trained on IoT datasets, old datasets are used instead. These old datasets do not contain records about new attacks or IoT-specific attacks. In this paper, we propose a novel intrusion detection system to detect attacks in protocol MQTT using Machine Learning. Before training the models, many techniques are used to pre-process the data and select the most important features from the dataset. The proposed algorithms are evaluated on a dataset named MQTT-IoT-IDS2020 that contains MQTT-specific attacks. Obtained results show that Decision tree and Logistic regression models outperform KNN and Naive Bayes models in terms of accuracy and precision.

Keywords:

IoT, Intrusion detection system, Machine Learning, MQTT, Feature

1. Introduction

IoT is an evolving technology that promises new dimensions of applications aiming to enhance and facilitate the quality of human life. However, the limited computing performance in addition to the significant number of connected devices in IoT along with their heterogeneity makes most of these resource- constrained devices unable to run sophisticated protocols effectively like conventional networks. Besides, IoT applications require different levels of QoS, so new communication patterns are required. Hence, new application layer protocols are rather adopted like MQTT, CoAP, DDS, AMQP [16].

MQTT is a lightweight real-time messaging protocol for IoT [9]. It is a widely used application layer protocol in IoT because it fits resource-constrained devices like IoT devices, also it is characterized by a reduced packet loss ratio and low bandwidth consumption [12]. MQTT is based on a publish/subscribe communication paradigm based on four key components: the broker, the IoT devices, a topic, and messages. All messages are exchanged between the IoT devices through the broker. Each IoT device is either a publisher, a subscriber, or both. The publisher is an IoT device, usually a sensor, that sends messages about a specific topic (ambient temperature, motion detection, ...). The subscriber is an IoT device interested in a specific topic like smartphones, workstations, etc. It receives messages from the publisher through the broker. Almost all exchanged messages have a 2 bytes protocol header, this small header makes the protocol suitable for constrained devices and machine-to-machine communications in IoT. According to [7], as the adoption of MQTT increases in IoT networks, it is critical to enhancing its security. Particularly, the publish/subscribe paradigm makes it more attractive for intruders which increases the number of possible attacks over MQTT. In [11], an analysis of MQTT and its attacks is presented. The authors classify the attacks into Man-in-the-Middle attacks, Denial of Service (DoS) attacks, and other intrusions. In Man-in-the middle attack scenario, the attacker either overhears the communication between the IoT devices or acts as one of the parties. Regarding the DoS attack in MQTT, it is usually conducted against brokers. However, intrusions refer to any other unauthorized activity in the IoT network. Besides, data privacy is another challenge in MQTT because by default data is transmitted in plain text, MQTT does not use any encryption technique which facilitates traffic sniffing. Further, the attacker who has already sniffed the traffic can alter the data in transit which threaten the data integrity. On the other hand, the manufacturers of IoT devices focus more on adding new functionalities to IoT devices, to make them smarter and more cost-effective. Unfortunately, the lack of security awareness makes them focus more on functionalities over security while a trade-off must be ensured. Thus, the

Manuscript received November 5, 2022
Manuscript revised November 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.11.110

devices come with inherited security vulnerabilities, hence becoming an attractive target for attackers [9].

According to recent studies [8] [13], machine learning-based intrusion detection systems have gained popularity due to their ability to produce excellent results in detecting malicious behavior. An ML-based IDS analyzes collected data to detect patterns that could indicate potential attacks on the target host or network. In IoT, host-based IDS are not recommended because of the limited resources and computing capability of the devices, network-based IDS fits more IoT environments. Compared with the extensive attention and research in Intrusion detection for IoT based on ML, there are few studies interested in attacks on specific IoT protocols like MQTT [14]. Further, existing approaches suffer from many limitations. First, the existing studies interested in MQTT attacks detection, use either the datasets collected from conventional IT networks or small-scale test beds [7]. Some of the used datasets are generated by computer simulations [5] [15]. Some studies seek to detect MQTT attacks, however, their models are based on the TCP protocol analysis, which does not provide enough details about the MQTT protocol [15]. Usually, the efficiency of the ML-based IDS highly depends on training the model using a suitable dataset. Similarly, to detect MQTT attacks in IoT, a dataset containing MQTT flow traffic and MQTT-specific attacks is required [8]. Secondly, many existing IDS approaches are based on very complex models that require a lot of time in processing and computing resources. Particularly, models based on Deep learning require high storage capacity, training time, and computational resources [4]. In IoT, a lightweight IDS is required to facilitate its deployment in real environments. Third, a way to reduce the complexity of the models is the dimensionality reduction of the dataset. Unfortunately, many research works interested on MQTT attack detection ignore this aspect [7]. In this paper, we present a new intrusion detection model for MQTT based on machine learning. We have trained four machine learning algorithms using a dataset named MQTT-IoT-IDS2020 [1]. Then, we conducted a set of experiments to evaluate the efficiency of our model in detecting attacks in terms of accuracy, precision, F1, and recall. The aim is to assess the capabilities of the proposed model in classifying the traffic into benign and malicious.

The key contribution of this paper is twofold:

- To reduce the complexity of the proposed model, the Random Forest built-in feature selection technique is used as a preliminary task to select the optimal set of features (in terms of classification) to be used in the training. Using such a technique has the potential to reduce the complexity of the model, also to reduce the generalization error. It provides an efficient and simple way of selecting features based on their importance in building the model and their impact on the target class.

- Unlike most existing studies interested in MQTT attack detection, our model is trained using an IoT dataset that contains MQTT traffic flows and MQTT-specific attacks. The dataset consists of five recorded scenarios: 1 benign operation scenario and four attack scenarios. The four considered attacks are as follows: Aggressive scan, UDP scan, Sparta SSH brute-force, and MQTT brute-force attack.

The remainder of the paper is structured as follows. Section II reviews similar approaches interested in MQTT attack detection. Section III describes the details of our methodology. Section IV presents the experiments details and results, and section concludes the paper.

2. Related Work

In this section, we will review and discuss recent approaches proposed to detect attacks in the MQTT protocol. A Deep Neural network model to detect intrusions in MQTT protocol is proposed in [4]. An extensive set of experiments has been conducted to compare many shallow algorithms to DNN. Results show that DNN outperforms shallow algorithms in terms of detection capabilities. Unfortunately, the complexity of the proposed model is very high compared to other algorithms. As they have shown in their paper, the training time for example bypasses 190 seconds compared to 8.9 seconds for DT which has roughly the same accuracy as DNN (about 90%). In [5], the authors propose an anomaly-based IDS for attack detection in MQTT using Machine learning. In their research work, they have built their own IoT dataset by simulating an IoT network. The created dataset is used to train many ML algorithms like Random Forest, Autoencoder, and K-means. As result, they have shown only the accuracy of the models which is not sufficient to give enough evidence of the performance of their models. In [6], the authors have trained many ML algorithms like Neural network, RF, naive Bayes, DT… using their own dataset named MQTTset. The authors claim that all existing IoT datasets are missing some aspects and/or are not IoT protocols enabled. That is why they have created a new dataset. They have collected data from a simulated IoT network where they have implemented many MQTT attacks like MQTT publish flood, flooding DoS, SlowITe, and brute force. The obtained results show that Neural Networks and RF outperform other tested models. Unfortunately, the authors did not compare their results to the existing IDSs model trained on another dataset to point out the efficiency of MQTTset. The authors propose in [7] a machine learning-based detection framework developed for the MQTT protocol to protect the MQTT brokers from DoS attacks. The detection process relies on many features based on the MQTT header and payload meta-data. To this end, they have developed their own testbed to create a dataset to be used later in training. They proposed three classifiers: AODE, Multi-Layer Perceptron, and decision tree. The proposed framework suffers from two limitations.

First, the testbed used to collect the dataset is very small. Besides, the framework is missing an efficient features selection technique. In [1], The authors have evaluated the effectiveness of six ML algorithms in detecting attacks in the MQTT protocol. The authors have implemented an IoT network based on MQTT protocol and many potential attacks. They have created a simulated dataset, used later to train different considered ML algorithms like the random forest, decision tree, k-NN, and SVM. Although the created dataset is interesting in terms of traffic and features diversity and the high scale of the considered network compared to existing models [6] [7], the experiment results show that the models do not have high detection capability. This may be explained by the missing of an efficient feature selection technique that needs to be applied prior to the training.

In [10], the authors designed a new dataset named SENMQTT-SET for DoS detection in MQTT protocol. The data was collected from a real testbed consisting of heterogeneous sensors and real-time devices. To extract the optimal set of features, an ensemble statistical multi-view cascade feature generation algorithm has been developed. Then, many ML algorithms have been evaluated to show the effectiveness and reliability of the proposed dataset. The best model has been deployed in a real network, and its performance has been evaluated using many metrics such as cumulative distribution, function, jitter, packet length, etc. Although the promising results, the considered testbed was very small compared to real IoT networks usually characterized by a high number of devices.

aggressive scan, UDP scan, Sparta SSH brute-force, and MQTT brute-force attack [1]. The network traffic corresponding to each of these scenarios is recorded in separate files for three abstraction- level network flow features of MQTT enabled simulated network. These flow features include Packet-flow, Uni-flow, and Bi-flow features. Every level of flow feature of MQTThas five files representing attacks and normal records of a particular scenario as mentioned above. In the current research,we are interested only in the uniflow traffic. Inspired bythe research work proposed in [4], we implemented a python script to combine all these five files of the uniflow level into one combined CSV. The combined CSV file contains the binary label attribute: normal or malicious, to test the ML algorithms over the MQTT protocol recorded traffic for binaryclassification.

Fig. 1 presents files of dataset MQTT-IoT-IDS2020 in each network flow feature and the combined version dataset of the uniflow set of features.

In the Uni-flow feature data of MQTT,there are five files, and we combined all of these five files intoone CSV with one extra column labeled 'attack'. In table I, we represent the list of the Uni-flow features.

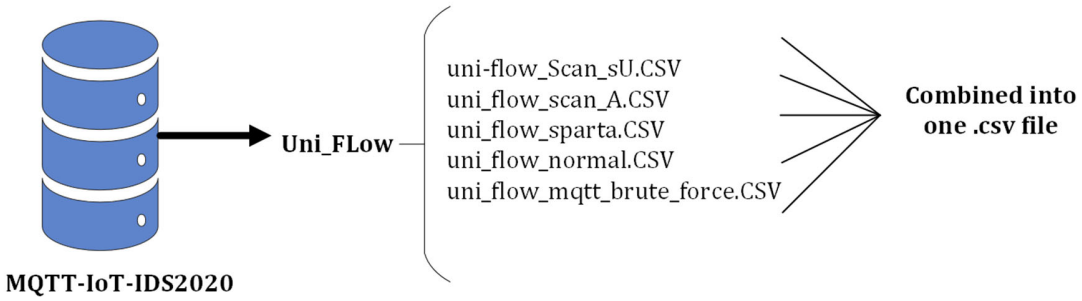

Fig. 1. The combined version dataset for uni flow-level features of MQTT-IoT-IDS2020

3. Methodology

3.1 Data collection

During the creation of the MQTT-IoT-IDS2020 dataset, five scenarios were launched: normal operation,

Table 1: Uni-flow features description

Feature	*Data type*	*Description*
proto	Integer	Transport Layer protocol
Mean_iat	Decimal	Average interarrival time
Std_iat	Decimal	Standard deviation of iat

Min_iat	Decimal	Minimum inter arrival time
Max_iat	Decimal	Maximum interarrival time
Numpkts	Integer	Number of Packets in the flow
Num_rst_flags	Integer	Number of reset flag
Num_bytes	Integer	Number of bytes
Num_psh flags	Integer	Number of push flag
Mean_pkt_len	Decimal	Average packet length
Std_pkt_len	Decimal	Standard packet length
Min_pkt_len	Decimal	Minimum packet length
Max_pkt_len	Decimal	Maximum packet length
Is_attack	Integer	1: attack and 0: normal

3.2 Data Analysis and Preprocessing

In this section, we will analyze the data to find any discrepancies, interesting patterns, correlations in data, etc. This step is popularly known as exploratory data analysis.

3.2.1 Data distribution:

According to the distribution of the target class shown in Fig.2, we notice that the considered dataset has

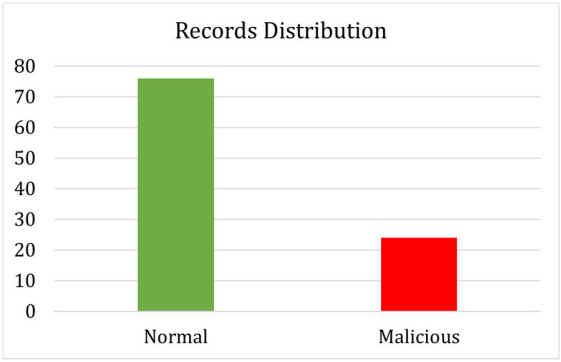

Fig 2. Target Class Distribution

two classes. The first class is labeled 'normal' which represents 76% of the total number of records and corresponds to benign traffic. The second class is labeled 'attack', it has 24% of the total number of records and corresponds to malicious traffic.

3.2.2 Correlation Heatmap

A correlation heatmap is a plot that visualizes the strength of relationships between numerical variables. Typically, it contains several numerical variables, where each variable is represented by a column. The rows represent the relationship between each pair of variables. The values in the cells indicate the strength of the relationship. A correlation heatmap can be used to find both linear and nonlinear relationships between variables. The correlation heatmap of the attributes of the Uni-flow dataset is shown in Fig.4.

It is obvious in Fig.4, that most of the data have a very low correlation. This is a crucial characteristic of the Machine Learning process. Besides few features have a high correlation with the target class namely num_pkts and num psh_flags. When two variables are highly correlated with each other, it's important to identify and remove one of them, hence we need to use a method for feature selection to select the most important features.

3.2.3 Scaling Numerical Attributes

Data Scaling and Normalization are common preprocessing techniques applied in Machine Learning, where the data is usually normalized to a scale of 0 to 1. While training Machine Learning algorithms, if the values of the features are closer to each other there is more chance for the algorithm to get trained better and faster. Consequently, the accuracy of the model increases compared to the case where the data or feature values have huge differences. Simply, the scaling is used to make the feature values generalized so that the distance between them will be lower. In the current study, we performed feature scaling to some features in Uni-flow data, particularly: mean_pkt_len, num_byte, min_pkt_len, max_pkt_len.

3.2.4 Feature Selection

Feature Selection is a preprocessing technique that selects the most "useful" features in the classification to reduce the complexity and the generalization error of the model and enhance the computational efficiency. It is applied when there is a potential for redundancy or irrelevancy of the features and to reduce the number of features usually known as features dimensionality reduction [22].

In the current research, the Random Forest feature selection model is used [22]. It is an embedded method that combines filter and wrapper methods. Random forest feature selection model is based on built-in feature selection methods characterized by high accuracy, better generalization, and interpretation compared to other feature selection methods.

	proto	num_pkts	mean_iat	std_iat	min_iat	max_iat	mean_pkt_len	num_bytes	num_psh_flags	num_rst_flags	std_pkt_len	min_pkt_len	max_pkt_len	is_attack
proto	100.0	-8.6	9.6	3.4	8.7	7.0	-13.0	-0.4	-10.2	-5.8	-14.6	8.5	-12.0	18.0
num_pkts	-8.6	100.0	0.9	16.1	-0.8	17.9	63.2	71.4	73.5	44.2	60.8	13.5	63.6	45.6
mean_iat	9.6	0.9	100.0	29.1	95.6	91.2	3.6	0.1	1.2	0.9	0.9	6.1	1.3	-0.5
std_iat	3.4	16.1	29.1	100.0	0.9	59.6	17.1	2.3	20.3	13.6	17.6	1.4	18.3	11.7
min_iat	8.7	-0.8	95.6	0.9	100.0	78.9	1.6	-0.1	-0.9	-0.5	-0.9	5.5	-0.6	-1.6
max_iat	7.0	17.9	91.2	59.6	78.9	100.0	16.0	2.2	19.9	11.6	14.9	4.6	15.8	14.7
mean_pkt_len	-13.0	63.2	3.6	17.1	1.6	16.0	100.0	23.7	72.6	71.5	89.0	30.2	90.2	6.9
num_bytes	-0.4	71.4	0.1	2.3	-0.1	2.2	23.7	100.0	10.5	8.3	10.2	31.7	12.3	5.6
num_psh_flags	-10.2	73.5	1.2	20.3	-0.9	19.9	72.6	10.5	100.0	52.5	81.3	-9.7	83.6	50.7
num_rst_flags	-5.8	44.2	0.9	13.6	-0.5	11.6	71.5	8.3	52.5	100.0	86.4	-20.1	86.4	31.4
std_pkt_len	-14.6	60.8	0.9	17.6	-0.9	14.9	89.0	10.2	81.3	86.4	100.0	-15.7	99.6	32.5
min_pkt_len	8.5	13.5	6.1	1.4	5.5	4.6	30.2	31.7	-9.7	-20.1	-15.7	100.0	-10.7	-46.5
max_pkt_len	-12.0	63.6	1.3	18.3	-0.6	15.8	90.2	12.3	83.6	86.4	99.6	-10.7	100.0	33.5
is_attack	18.0	45.6	-0.5	11.7	-1.6	14.7	6.9	5.6	50.7	31.4	32.5	-46.5	33.5	100.0

Fig 3. Correlation Heatmap

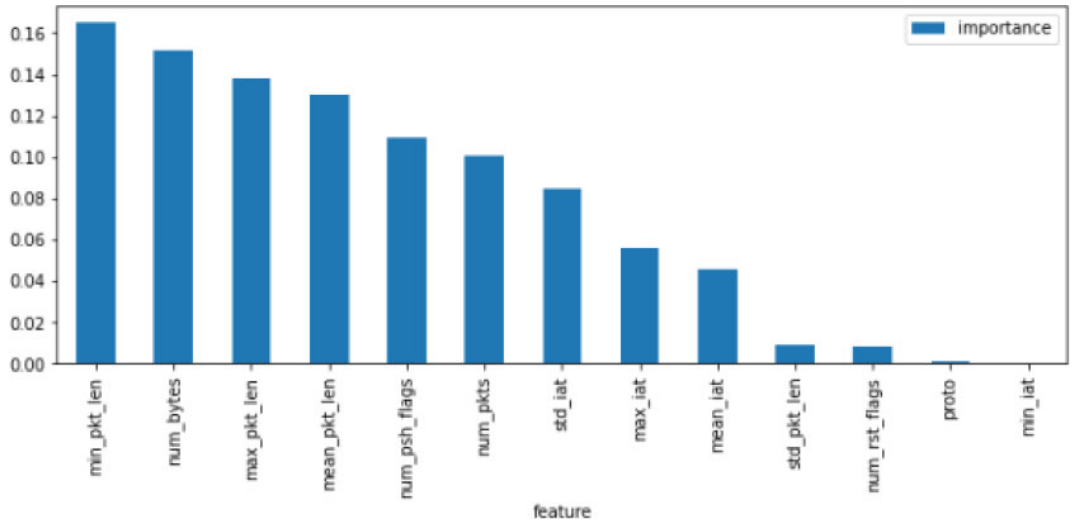

Fig. 4. Visualization of the features' importance.

Random forest consists of many decision trees, each of which is built over a random extraction of the observations from the dataset and a random selection of the features. Each tree is a sequence of nodes that consists of yes-no questions based on a single or combination of features. At each node, the tree divides the dataset into 2 branches, each of which hosts similar observations different from the ones in the other branch. Therefore, the importance of each feature is derived from how "pure" each of the branches is [22].

Random forest is applied to the Uni-flow dataset based on the following steps. First, we specified the random forest classifier instance. Then, we use the recursive feature elimination (RFE) from the 'sklearn' library to automatically select the features whose importance is greater than the mean importance among all other features.

Let us note here that in our scenario we use a threshold of 0.01 for the importance. It is important to mention that in all feature selection procedures, it is a good practice to select the features by examining only the training set to avoid over-fitting. Hence, we select the features from the train set and then transfer the changes to the test set later. Fig. 4 shows the distribution of the features selected using the above-described technique.

3.2.5 Modeling

As shown in Fig.5, which presents the different steps of our methodology, the prepossessed dataset is split into two parts 70% for the training and 30% for the testing

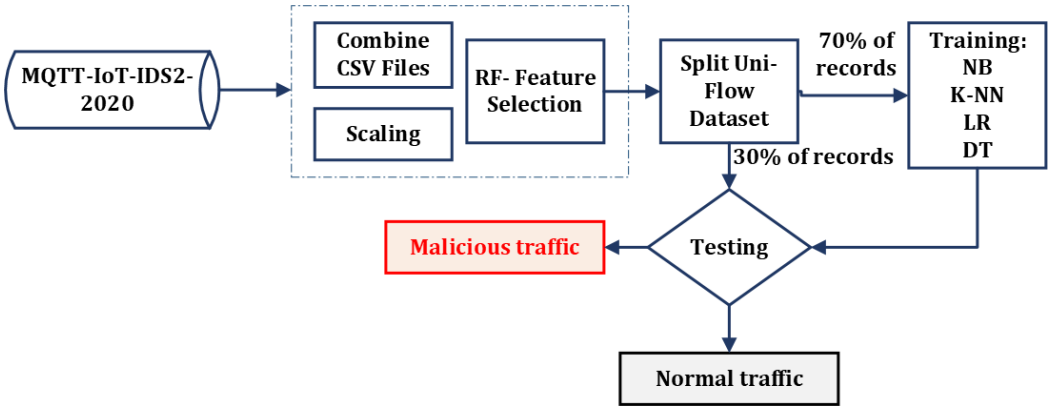

Fig 5. Methodology

of the model.

The following phase is training, where four algorithms are considered: Naive Bayes (NB), Logistic Regression (LR), k- Nearest Neighbour (kNN), and Decision Tree (DT). In the following each of which is described:

- Naive Bayes (NB): It is a probabilistic classifier based on the Bayes Theorem. It is a scalable algorithm that does not require huge training datasets to produce appreciable results. The na¨ıve Bayes classifier assumes that the occur- rence of a certain feature is independent of the occurrence of other features [8].
- Logistic Regression (LR): Logistic Regression (LR) is a supervised ML algorithm that is primarily used in binary and multiclass classification. The LR model applies the sigmoid function or its variations to a linear ML model to ensure that the output is in the interval [0,1]. It is a predictive analysis algorithm based on probability. The sigmoid function is used to map the predictions to probabilities [17].
- k-Nearest Neighbor (k-NN): k-NN is one of the simplest supervised ML algorithms which relies on the similarity of the features to predict the class of a given data sample. It identifies a sample based on its Euclidean distance to its neighbors. In the K-NN algorithm, k is the number of nearest neighbors used for classification. The performance of the model highly depends on k. If the value of k is very small, the model may be susceptible to over- fitting. however, a large value of k value may result in misclassification of the sample. The K-NN technique has the advantage of being an analytically tractable classifier for IDSs [8].
- Decision Tree (DT): DT is one of the basic supervised ML algorithms which is used for both classification and regression of the given dataset by applying the series of decisions (rules). The model has a conventional tree structure with nodes, branches, and leaves. Each node represents a feature. The branch represents a decision while each leaf represents a class label. The DT algorithm automatically selects the best features for building a tree and then performs pruning operations to remove irrelevant branches from the tree to avoid over-fitting [8].

4. EXPERIMENTS AND RESULTS

In all experiments, training and testing were conducted on the Google Colaboratory Pro platform. To evaluate the performance of the classifiers, we use the commonly used metrics: accuracy, precision, recall, and F1 score. They are calculated based on the percentage of true negatives (TN), falsepositives (FP), false negatives (FN), and true positives(TP) as follows:

- Accuracy: It measures how many observations, both positive and negative, were correctly classified, it is computed as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

- Precision: is the ratio between the true positives and allthe positives.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

- Recall: it is the proportion of actual positives which are predicted positive.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

- F1: a measure that combines precision and recall.

$$F1=2*\frac{precision \times recall}{precision+ recall} \quad (4)$$

Figures 6, 7, 8, and 9 plots the accuracy, precision, recall and F1 score, respectively for the model. To compare the obtained results to existing approaches, we also present the performance of the models proposed in [1].

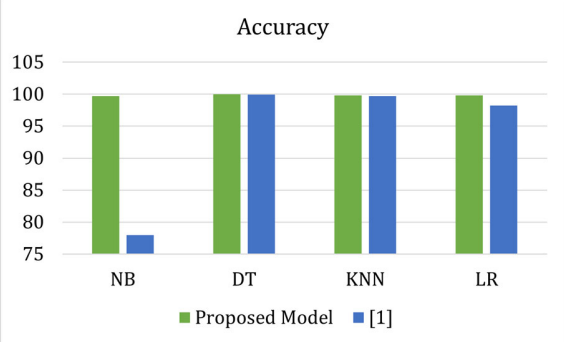

Fig. 6. Accuracy of the different models

Figure 6 shows that in our model the NB and LR have an accuracy of 99.7% and 99.8% respectively compared to the performance of the model proposed in [1] where NB and LR have an accuracy of 78% and 98.23%. Whereas DT and KNN have close performance, the accuracy is about 100%.

As depicted in Figures 7, 8, and 9, the precision, F1 and recall of our classifiers outperform the models proposed in [1]. For instance, the precision of our classifiers is about 99% compared to the classifiers proposed in [1]. This result is due to the RF feature selection technique used in our model, particularly, the selection of the most optimal set of features before training,

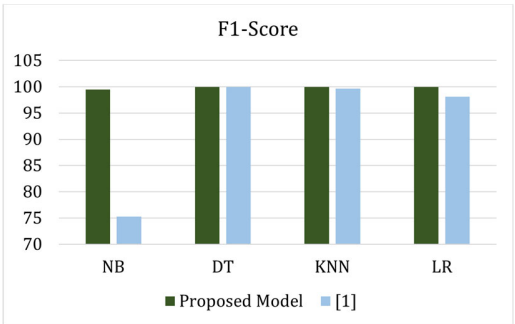

Fig. 9. Recall of the different models

makes the model be more deterministic and reduce overlapping. Besides, the highest accuracy (99.97%) and F1-score (100%) are obtained for DT, it has the best

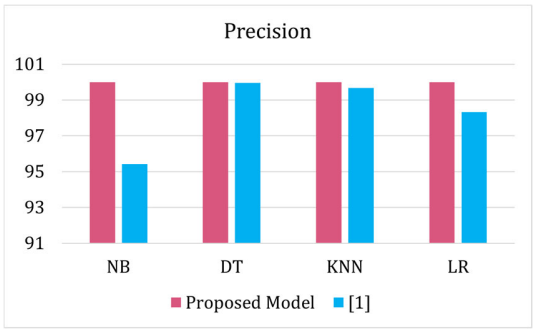

Fig. 7. Precision of the different models

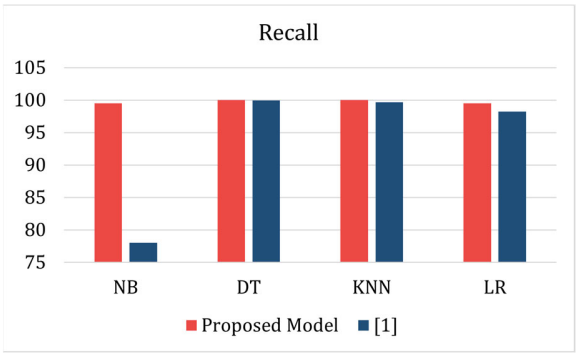

Fig. 8. F1 of the different models

performances among all the four classifiers, k-NN has close performance to DT.

3. Conclusion

In this paper, we present an ML-based intrusion detection system for MQTT IoT smart systems. In the training phase, we use a recent dataset named MQTT-IoT-IDS2020 which contains records about benign and malicious traffic in an IoT environment. A pre-processing phase precedes the training of the algorithm where we normalized and scaled data in addition to a feature selection phase where we selected the most important features of the dataset to enhance the accuracy of the algorithm. The obtained results show that, for Uni-flow data, the highest accuracy is achieved with DT. Additionally, our classifiers outperform the existing models trained on the same dataset. As future work, we expect to use a more sophisticated feature selection technique and explore other machine learning algorithms.

References

[1] Hanan Hindy, Ethan Bayne, Miroslav Bures, Robert Atkinson, Christos Tachtatzis and Xavier Bellekens. (2020), Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset).

[2] Hindy, H., Tachtatzis, C., Atkinson, R., Bayne, E., Bellekens, X.: MQTT- IOTIDS2020: MQTT internet of things intrusion detection dataset. IEEE Dataport (2020). https://doi.org/10.21227/bhxy-ep04
[3] Akash Dubey., on Dec 15, 2018, Feature Selection Using Random forest. Towards Data Science TDS.
[4] M.A Khan, , A.K. Muazzam, J. Sana Ullah, A. Jawad, J. Sajjad Shaukat,
[5] S. Awais Aziz , P. Nikolaos, and J.B. William 2021. "A Deep Learning- Based Intrusion Detection System for MQTT Enabled IoT" Sensors 21, no. 21: 7016. https://doi.org/10.3390/s21217016.
[6] E. Ciklabakkal, A. Do¨nmez, M. Erdemir, E. Suren, M. T. YILMAZ, and
[7] P. Angın, "ARTEMIS: An intrusion detection system for MQTT attacks in Internet of things," 2019, Accessed: 00, 2020.
[8] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, E. Cambiaso, MQTTset, a New Dataset for Machine Learning Techniques on MQTT, Sensors (Basel, Switzerland), vol. 20,22 6578. 18 Nov. 2020.
[9] N.F. Syed, Z. Baig, A. Ibrahim, C. Valli (2020) Denial of service attack detection through machine learning for the IoT, Journal of Information and Telecommunication, 4:4, 482-503.
[10] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, A. Wahab. 2020. "A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions" Electronics 9, no. 7: 1177.
[11] M.Husnain, K. Hayat, E. Cambiaso, U.U. Fayyaz, M.Mongelli, H. Akram, S. Ghazanfar Abbas, G.A. Shah, Preventing MQTT Vulnera- bilities Using IoT-Enabled Intrusion Detection System. Sensors 2022, 22, 567. https://doi.org/10.3390/22020567.
[12] S. Hariprasad, T. Deepa, P. Chandhar, SENMQTT-SET: An Intelligent Intrusion Detection in IoT-MQTT Networks Using Ensemble Multi Cascade Features, IEEE Access, vol. 10, pp. 33095 - 33110, 2022.
[13] A.P. Singh, A. Kumar, V. Kumar, A Study on MQTT Protocol and its Cyber Attacks. International Advanced Research Journal in Science, Engineering and Technology, Vol. 9, Issue 1, January 2022. DOI: 10.17148/IARJSET.2022.9136.
[14] S. Andy, B. Rahardjo and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," 2017 4th In-ternational Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2017, pp. 1-6, doi: 10.1109/EECSI.2017.8239179.
[15] Zhiyan Chen and Jinxin Liu and Yu Shen and Murat Simsek and Burak Kantarci and Hussein T. Mouftah and Petar Djukic, Machine Learning- Enabled IoT Security: Open Issues and Challenges Under Advanced Persistent Threats. ACM Computing Surveys. April 2022.
[16] F.B. Saghezchi, G. Mantas, M.A.Violas, A.M. de Oliveira Duarte, J. Rodriguez, Machine Learning for DDoS Attack Detection in Industry
[17] 4.0 CPPSs. Electronics 2022, 11, 602.
[18] N. Moustafa, B.P. Turnbull, K. Choo, (2019). An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things. IEEE Internet of Things Journal, 6, 4815-4830.
[19] D.Silva, L.I. Carvalho, J. Soares, R.C.Sofia, A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA. Appl. Sci. 2021, 11, 4879.
[20] F. Abbasi, M. Naderan, S.E. Alavi. (2021). Intrusion detection in IoT with Logistic Regression and Artificial Neural Network: further inves- tigations on N-BaIoT dataset devices. 10.22108/jcs.2021.129807.1077.
[21] N. Pudjihartono, T. Fadason. (2022). A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. Frontiers in Bioinformatics.https://doi.org/10.3389/fbinf.2022.927312
[22] Y. Akhiat, Y. Manzali, M. Chahhou, A. Zinedine. (2021). A New Noisy Random Forest Based Method for Feature Selection. Cybernetics and Information Technologies. 21. 10-28. 10.2478/cait-2021-0016.

Tahani Gazdar received the Engineering, M.Sc., and Ph.D. degrees in computer science from the National School of Computer Sciences, University of Manouba, Tunisia, in 2009, 2010, and 2015, respectively. She is currently an Assistant Professor with the College of Computer Science and Engineering, University of Jeddah, Saudi Arabia. Her current research interests include trust management, blockchain and ML application in Cybersecurity.