Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7
© 2022 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DENIAL OF SERVICE CLASSIFICATION ON MESSAGE QUEUEING TELEMETRY TRANSPORT PROTOCOL AT INDONESIA OIL SERVICES COMPANY

[1]GRENDY E ALIANDY, [2]RIYANTO JAYADI

[1,2]Information Systems Management Department,

BINUS Graduate Program - Master of Information Systems Management,

Bina Nusantara University, Jakarta, Indonesia 11480

E-mail: [1]grendy.aliandy@binus.ac.id, [2]riyanto.jayadi@binus.edu

ABSTRACT

In Indonesia, adopting the Internet of Things (IoT) platform, which employs the Message Queuing Telemetry Transport (MQTT) protocol as a communication system, is growing. Most IoT systems lack adequate security mechanisms, and MQTT is vulnerable to various attack scenarios. Despite these security mechanisms being inefficient for low-power devices, MQTT's standard security is SSL/TLS. DoS attacks, which are a sort of MQTT protocol attack, have the potential to damage the entire IoT infrastructure. This attack can cause serious effects, such as data transmission interruptions, server inaccessibility, and tool monitoring in real-time. Based on data from attacks on IoT systems in Indonesian oil services organizations and data from prior studies, this research aims to find the most accurate machine learning system. This method assumes that a stack of machine learning algorithms can efficiently detect DoS attacks. According to the findings, the stacked ensemble approach has a 99.6489 percent accuracy rate in identifying attacks. We present a systematic approach for detecting attacks on IoT platforms in this study. This paper's approach is intended to serve as a standard for detecting DoS in MQTT-based IoT systems.

Keywords: *IoT, MQTT, Machine learning, DoS attack, Stacked ensemble*

1. INTRODUCTION

Industry 4.0, the IoT, and digitization are all about gathering data, making it available, analyzing it, and leveraging it to create value by enhancing productivity through data availability [1]. The first stage, most importantly, is to make the data available to the applications that require it. We might install these applications on a machine except the one that generates the data or even cloud-based. As a result, one of the most critical aspects of digitization is transferring data from its source to its intended purpose [2]. A system of interconnected computing devices, animals, objects, mechanical and digital machines or people is provided with unique identifiers (UIDs), and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction is how the Internet of Things (IoT) is defined [3].

Nowadays, the IoT is transforming into a powerful technology that IoT will widely use to supplement or perhaps replace many other technologies [4]. Some of the most common IoT applications are healthcare, intelligent automobiles, and smart cities. Smart cities and intelligent households can save up to 20% on energy costs, but they also have a slew of security vulnerabilities with their intelligent equipment [5]. IoT monitoring and control are simple to perform using smartphones and browsers [6]. Because of that, IoT is quickly expanding due to its potential advances in increasing comfort and efficiency in everyday life use. According to Gartner, installed IoT use will reach 5.81 billion in 2020 [7]. Aside from that, [8] illustrates that by 2022, Indonesia is expected to have 400 million IoT devices.

Industry can use IoT technology to digitize data by transmitting it from the source to the cloud via various communications protocols. Several Internet of things (IoT) applications have been created and deployed [9]. There are several communications protocol alternatives available now for IoT devices. HTTP, SMPP, AMQP, CoAP, and MQTT are five of the most well-known IoT protocols [10].

Furthermore, several factors should be considered when choosing the most effective message protocol, including energy efficiency, performance, reliability, and resource utilization [11]. The MQTT protocol is one of the better alternatives for functionality, reliability, and the ability to obtain multicast communications [12]. MQTT is the most frequently used protocol, yet it has the lowest security among the MQTT protocols. Based on this analysis, we must enhance the MQTT protocol's security by understanding the types of attacks that might occur. According to [13], although the MQTT protocol is the most widely used, it has the least secure security of the MQTT protocols. Based on these findings, we should improve the MQTT protocol's security to survive possible attacks.

2. RELATED WORKS

Study [14] designed numerous attack scenarios on the MQTT protocol to explore what kind of attacks are possible. We can minimize the known types of attacks for prevention. Study [15] performed multiclass classification to detect attacks on the MQTT-IoT protocol. DoS attacks are one of the most severe security risks to be considered [16]. The study [17] created a lightweight security mechanism that handles DoS attacks on the MQTT protocol. Study [18] presents various security threats and issues at multiple layers of IoT application: sensing layer, middleware layer, gateway, and application layer, and find open problems that stem from the solution itself. Another work suggests an IoT surveillance system with sensor nodes and actuators exchanging data via a secure MQTT protocol[19].

The efficacy of six machine learning algorithms for detecting MQTT-based risks was tested in the study[20]. By comparing six machine learning algorithms, it was discovered that one-way and two-way features are better suitable for discriminating between benign and MQTT-based attacks for categorization. The study [21] proposed a system for detecting MQTT DoS attacks based on machine learning. The research technique began with defining a DoS attack model, followed by developing a framework that included a feature extraction engine, a network traffic generator, and a machine learning-based DoS attack traffic classifier. The study[22] MQTT dataset focusing on the MQTT protocol was presented, validated by a hypothetical detection system, and compared to a legitimate dataset including cyberattacks against the MQTT network.

Based on this information, the security of the MQTT protocol must be improved by understanding the types of attacks that can occur. Then, in [23], various attack scenarios on the MQTT protocol were created to determine the types of attacks conducted on the protocol. Mitigation can be done for prevention based on known types of attacks, and [15] performed multiclass classification to detect attacks on the MQTT protocol. MQTT is a popular protocol for IoT and Industry 4.0 applications. It's an interesting fit for many of these applications. However, the protocol has some limitations that we should recognize, including only "fire and forget" applications, quality of service feature not adequately described and not used. Actual queuing methods may be better suited for specific applications. Despite this, the MQTT lacks a comprehensive security approach because it has an authentication method and no encryption capabilities. Apart from that, many IoT systems lack sufficient security mechanisms, and MQTT may be readily exploited through several attack scenarios, one of which is a denial-of-service attack[14]. In today's IoT platforms, MQTT is frequently utilized as a message exchange protocol[24].

In general, an IoT platform provider with an IoT platform as a service business strategy. IoT Platform as a Service (Paas) is a cloud-based computing paradigm that allows development teams to build, test, deploy, manage, update, and expand applications more quickly and cost-effectively. In this research, an oil services company in Indonesia provides an IoT platform to several customers. This company's business model offers a solution to a final customer without hosting it on-premises, with sophisticated implementations and high overhead. It gives middleware and runtime to the user/customer, which solely manages data and applications and virtualization, storage, network, and servers.

An IoT communication protocol, MQTT, is used to design the platform and communication across microservices. MQTT is also used to transport data from devices to the IoT platform. MQTT was adopted as the protocol of choice since it is a lightweight protocol that can connect hundreds of IoT devices. Every day, it can handle about 50000 data packets. An attack was launched against this protocol due to its public availability, resulting in reduced platform performance and even the platform's inaccessibility. After we performed the investigation, we revealed that a DoS attack had been carried out, flooding the MQTT broker with data. To maintain the platform's performance and the SLA agreed upon with the customer, this company requires descriptive analysis to identify and mitigate attacks on the MQTT protocol in real-time. Earlier studies on identifying MQTT attacks have all relied on datasets supplied by libraries, such as those generated by [22] or simulations [15]. Furthermore, they focus on techniques that can accurately identify attacks but do not describe how to implement them on existing systems.

This work has the following major contributions: (1) This study focuses on five machine learning algorithms for detecting DOS attacks on the MQTT protocol utilizing data generated by [15] and actual data from an Indonesian oil service company's IoT system; (2) this study will propose a design framework for detecting DOS attacks on MQTT using the generated model; (3) formal analysis of attack classification with the dataset from the existing system; and (4) numerous simulations are performed to compare attack classification with other algorithms and compared to the previous study.

3. METHODS AND MATERIALS

Data Mining is a process of finding a relationship that means a tendency to examine a set of big data is stored in the storage with the technical introduction of patterns such as engineering statistics and mathematics. Data mining contains multiple processes to find hidden information in the study [25]. According to various surveys and user polls, this study uses CRISP-DM as research methodology because it is still the de facto standard for constructing data mining and knowledge discovery applications [26].

3.1 Data Understanding

Based on experience and qualified assumptions during this phase, hypotheses about concealed information about the data mining project's objectives are generated. Its primary goal is to become familiar with the data, identify exciting subgroups, and gain preliminary insights. Identify relevant information from a variety of databases that already exist. Several critical points should be considered in the data identification and (data) segregation procedure. Several key points should be considered in the data identification process and (data) selection phase.

This phase aims to discover any issues with the data's quality. The subset in charge of this step uses the data to form an initial hypothesis. According to this research, information protocol decision-making is critical in communication on IoT devices that don't have information, examining attacks on the MQTT IoT protocol.

3.1.1. Source of dataset

The dataset used in this study is from research in [15] and an IoT system environment in the Indonesian oil company. The first dataset consisted of 45514 attack data and 49111 normal data. The second dataset from the actual system, as shown in Figure 1.

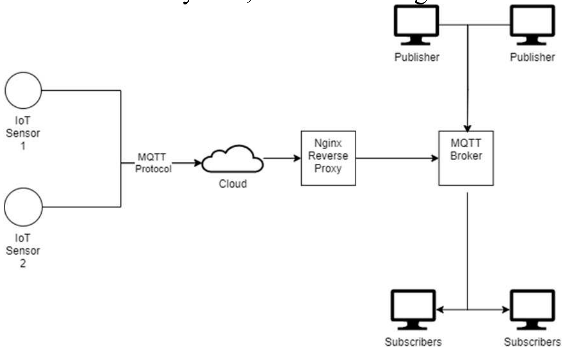

Figure 1: IoT System Environment

The acquired dataset contains 75740 attacks and 66856 normal data from the system's transactions. The DoS attack is used in both datasets. One of the most popular attacks on the Internet is the DoS attacks [27]. Every second, the broker is saturated with many new connections and messages in the MQTT protocol. In [15] research, the MQTT-malaria tool[28] is used to validate MQTT system scalability and load testing utilities.

This study uses the rStudio tools with R language to perform exploratory data analysis to understand better the distribution of data and the proportion of data. The [15] dataset has 94625 observations and 68 variables, while the IoT platform dataset has 142596 observations and six variables, according to the information available from the two datasets. In all datasets, the response variable "Type" is a binary category variable that includes DoS and Normal values. In addition to the column message, we can read it and take a quick look at our data by creating a new column called Datetime (convert from timestamp column) and converting all strings to factors to train the model quickly.

There are 67 predictor variables in the dataset [15], seven predictor variables in the IoT platform's dataset, and a target variable called "Type". It tracks 67 and 7 critical parameters across various topics (time delta, time epoch, time relative, ip src, MQTT msg, etc.).

3.2 Data Preparation

We define the distribution of attack and normal data from the dataset. According to the data distribution, it is discovered that attacks on the MQTT protocol often occur in minutes. Columns containing most NA data can be eliminated and not included in modeling. Based on the exploration results, the columns selected are frames.time_delta, frame.time_epoch, type due to the column strongly correlates with the attack / normal classification. Furthermore, the column will be explored and get the results. The correlation between the columns is examined to determine whether a column may be included in the modeling process. The standard deviation is zero since the data in columns 'mqtt.clientid','mqtt.msg','mqtt.topic', 'type', and 'datetime' are not numeric.

To validate the model we've built, we need to split the dataset into three parts: data train, data validation, and data test. Pre-split training (60%), validation (20%), and test (20%) datasets were used to split the dataset. To build classification models, we may directly use these two datasets as our training and test sets. We'll do the latter in this study because it will allow us to split the data into training, validation, and test sets and provide us greater flexibility in selecting the split percentages. In addition, missing values were eliminated from the column and row before training the model. We checked the percent of data between attack and normal to see if upsample or downsample required in modeling. The dataset [15] comprises 94625 observations after deleting rows with missing values, resulting in a 21 predictor with 51.8 percent positive observations and 48.2 percent negative observations, and IoT platform's dataset comprises 142596 observations after deleting rows with missing values, resulting in a seven predictor with 53.115 percent positive observations and 46.885 percent negative observations

To train the model, we divide the datasets [15] into two datasets, i.e., train dataset, which is a dataset that ignores NA data and novar dataset, which is the dataset that eliminates train data that has no variance. On the other side, the dataset used on the IoT platform is all variables(columns).

3.3 Classification Methods

This phase involves creating a data mining workflow to find the appropriate parameter settings for the chosen algorithm and performing data mining operations on previously processed data. We chose H2O because it allows us to train with multiple algorithms simultaneously, supporting R language. The H2O AutoML algorithm relies on quick training of H2O machine learning algorithms to construct many models in a short amount of time [29].

H2O is an open-source, distributed, rapid, in-memory, scalable machine learning and prediction platform that enables us to quickly develop machine learning models based on the datasets we use. To access and reference data, objects, models, and other stuff in H2O, the Distributed Key/Value Store is used across all nodes and machines. This method is based on the H2O distributed Map/Reduce Framework, which uses the Java Fork/Join architecture to be distributed and multi-threaded. Data is read parallel and spread throughout the cluster before being compressed and stored in memory in column format.

3.3.1. Dataset Splitting

The dataset must be divided into train data, validation data, and test data to validate the developed model. The dataset was split using pre-split training data sharing (60 percent), validation (20 percent), and test (20 percent). Validation data is required to ensure that the created model is not overfitting. When creating the classification model, both datasets can be evaluated and tested according to the training model. The dataset is split to allow greater freedom in determining the split percentage.

3.3.2. Algorithms Selection

The selection of machine learning algorithms is based on the previous study [15]. The study used Random Forest, GBM, and Deep Learning algorithms. The algorithm recommended by H2O auto ml was tested to find the most accurate algorithm in this study, as shown in Table 1. In addition, the stacked ensemble will be tested to produce better accuracy based on the results of research [30] which provides a recommendation to combine base-tier into a stacked ensemble algorithm to improvise accuracy.

Table 1: Best five results from H2O's automl

model_id	***auc***	***logloss***	***aucpr***
StackedEnsemble_BestOfFamily_AutoML_20210925_191343	1	0.0002	1
GLM_1_AutoML_20210925_191343	1	0.0002	1
GBM_1_AutoML_20210925_191343	1	0.0002	1
GBM_2_AutoML_20210925_191343	1	0.0002	1
GBM_3_AutoML_20210925_191343	1	0.0002	1

3.3.3. Generalized Linear Model

The Logistics Regression model with hyperparameters was employed for the first time in this study. The random hyperparameters search feature on the H2O learning machine is used to get hyperparameter values. After obtaining the value of the most optimal hyperparameters, these values will be used to train machine learning on the three datasets indicated in the preceding point. A grid search is one method for determining hyperparameter combinations. The Mean Cross Validation is a metric for evaluating the grid search's performance (CV). The values entered in the hyperparameters are combined in a grid search. Here are the values for the hyperparameters that were used:

1. Alpha is a set of numbers ranging from 0 to 1 in 0.001 increments.
2. Lambda is a set of numbers ranging from 0 to 1 in 0.000001 increments.

For the grid search, the following search criteria were used:

1. Strategi = *RandomDiscrete*
2. max_runtime_secs = 10*3600,
3. max_models = 100,
4. stopping_metric = "AUC",
5. stopping_tolerance = 0.00001,
6. stopping_rounds = 5,
7. seed = 1234

The generalized linear model (glm) approach will train machine learning with these parameters. The datasets used are training, and validation datasets from the three different datasets obtained.

3.3.4. Distributed Random Forest

We developed a distributed random forest model using random hyperparameter search and used the three datasets for the second model. The parameter values for the best Random Forest model and the AUC value and ROC curve for the validation dataset are shown in the output below. For the best Random Forest model, the following hyperparameters are used: ntrees = 10000 (early stopping), max depth = 15, min rows = 10, nbins = 30, nbins cats = 64, mtries = 2, sample rate = 1.

The ROC curve can be used to calculate the trade-off between the model's sensitivity (True Positive) and specificity (False Positive) (False Positive Rate). In the output area of the evaluation section, the variable importance is also provided. Factors like eth.src, according to the results, have limited predictive power. While utilizing the novar dataset, other characteristics such as ip.dst and eth.dst were more influential in deciding whether an MQTT message was classed as an attack.

3.3.5. Gradient Boosting Model

The gradient boosting machine with hyperparameters is the next model to be evaluated. The random hyperparameters search feature on the H2O learning machine is used to get hyperparameter values. The method used to find hyperparameter values is the same as in the prior models. Here are the values for the hyperparameters that were used:

1. learn_rate = seq(0.01, 0.1, 0.01),
2. learn_rate_annealing = seq(0.1, 1, 0.1),
3. max_depth = seq(2, 10, 1),
4. sample_rate = seq(0.5, 1.0, 0.1),
5. col_sample_rate = seq(0.1, 1.0, 0.1)

3.3.6. Neural Network

A random hyperparameter search was also used to select the Neural Network model. The Neural Network (Deep Learning) model parameter values are: epoch = 200, annealing rate 1e-8, rho = 0.999, epsilon = 1e-4, dropout = 0.2, activation = (Tanh, TanhWithDropout, Rectifier , RectifierWithDropout, Maxout, MaxoutWithDropout) and will be evaluated using the AUC value and ROC curve for dataset validation. The method used to find hyperparameter values is the same as in the prior models. Here are the values for the hyperparameters that were used:

1. hidden = list(c(5, 5, 5, 5, 5), c(10, 10, 10, 10), c(50, 50, 50), c(100, 100, 100), c(200, 200)),
2. epochs = c(50, 100, 200),
3. l1 = c(0, 0.00001, 0.0001),
4. l2 = c(0, 0.00001, 0.0001),
5. rho = c(0.9, 0.95, 0.99, 0.999),
6. epsilon = c(1e-10, 1e-8, 1e-6, 1e-4),
7. momentum_start = c(0, 0.5),
8. momentum_stable = c(0.99, 0.5, 0)

3.3.7. Stacked Ensemble

A stacking ensemble model was also built in this study by combining four base models: Logistic Regression, Random Forest, Gradient Boosting Machine, and Neural Network. We use hyperparameters values from the best models previously chosen.

3.4. Evaluation Metrics

The Confusion Matrix is used as a reference in this study to evaluate the algorithm performance of Machine Learning (particularly supervised learning). The Confusion Matrix depicts the data generated by machine learning algorithms' predictions and actual conditions [31]. We can determine Accuracy, Precision, and Specificity using the Confusion Matrix. Because each metric measures efficiency differently, we use a combination of metrics in our model to provide a more accurate picture.

3.4.1. Accuracy

Accuracy is the percentage of True (positive and negative) predictions concerning the total data. We can answer the question "What percentage of correct messages predict attacks and normal for all messages" using accuracy.

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

3.4.2. Precision

It's the percentage of true positive predictions to overall positive expected results. The question "What percentage of the accurate messages in the predicted attack of the total messages predicted attack?" can be answered using precision.

$$Precission = \frac{(TP)}{(TP + FP)} \qquad (2)$$

3.4.3. Specificity

It is the accuracy of predicting negative data compared to all negative data. We can answer the question "What percentage of correct messages is predicted to be normal relative to the overall message that is normal?" using specificity.

$$Specificity = \frac{(TN)}{(TN + FP)} \qquad (3)$$

3.4.4. F1 Score

The F1 Score is a combination of precision and recall that is weighted.

$$F1\ Score = 2 * \frac{(Recall * Precission)}{(Recall + Precission)} \qquad (4)$$

3.4.5. AUC – ROC

When classifying, AUC metrics perform very well [32]. The AUC-ROC curve is a performance metric that can identify problems at different threshold levels. The ROC is a probability curve, but the AUC is a separability measurement. AUC determines the model's ability to discriminate between classes.

True Positives are preferred in this study, and False Positives are carefully avoided because we choose a message that is an attack but is not predicted to be an attack, besides a message that is not an attack but is predicted to be an attack. Thus we tune to achieve high accuracy, specificity and precision levels.

4. RESULT AND DISCUSSION

The data mining findings are compared to the underlying business objectives, and the trained model is verified against real data sets on the IoT platform. A predictive study will conclude an attack trend that will occur in the future based on existing historical data. Machine learning will collect and create variables from the attack data in a single simulation. Additionally, the outcomes of this machine learning analysis will be improved by predictive analytics in the form of recommendations for items that may be utilized to predict the future.

The delta time and most topic variables are highly influential in detecting attacks using glm models, as shown in Figure 2. In contrast, as shown in Figure 3, the model with the novar dataset that eliminates the subject variable tends to make the IP variable (both destination and source) a variable that plays a crucial role in attack detection. As a result, the model with the novar dataset has lower accuracy and AUC than the model with the alltrain dataset.

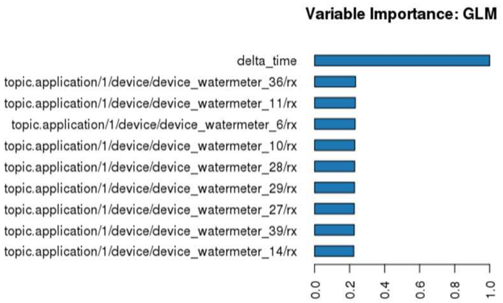

Figure 2: Variable importance on alltrain dataset

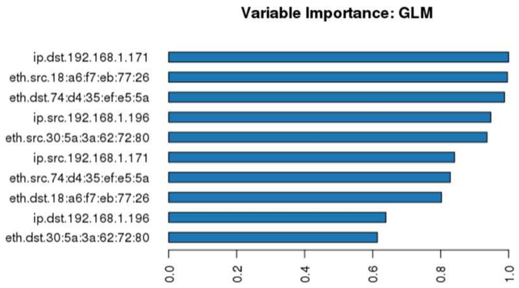

Figure 3: Variable importance on novar dataset

The AUC criterion was used to evaluate all classification models in this study. The parameter values for the best Logistic Regression model in the IoT platform's dataset with seven predictors and the AUC value and ROC curve for the validation dataset and variable coefficients Table 3. The hyperparameter values for the optimal Logistic Regression model are alpha = 0.144 and lambda = 0.005041, respectively. Furthermore, the validation dataset's AUC score is 0.9272472.

H2O system determines that the columns/predictors' frame. offset shift', 'frame.marked', 'frame.encap', 'type', and 'frame.omitted' in the dataset [15] and predictors' time', 'retain', from IoT platform's dataset can be ignored from the entire train dataset since they include relatively stable values. The AUC value for the four base models and the ensemble model from the IoT platform's dataset is displayed in figure 2. In addition, the ensemble model's coefficients are included in the result. According to the AUC values, the Ensemble model has the highest prediction accuracy with a value of 0.9981467.

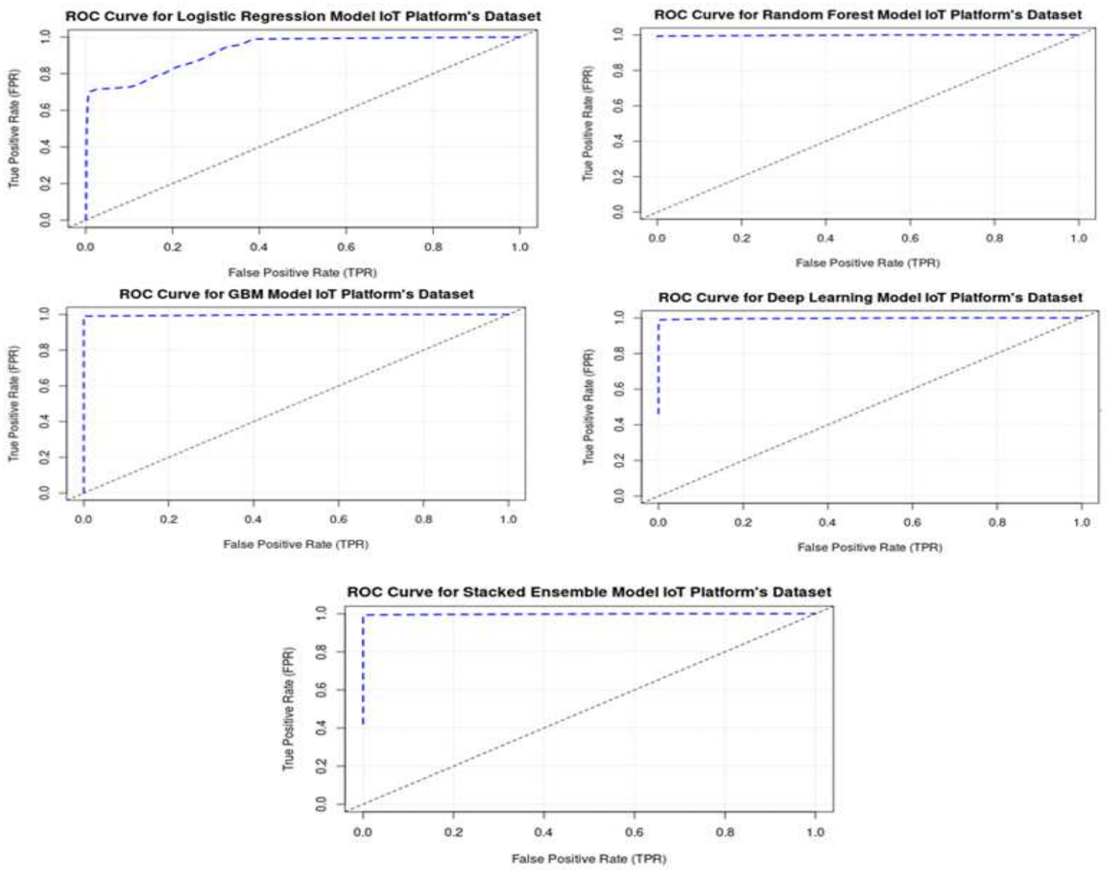

Figure 4: ROC Curve for IoT Platform's dataset

The ensemble model md_ens was chosen as the best model for prediction after analyzing the AUC values of various models. As illustrated in figure 2, we can obtain the AUC value and ROC curve of the best model for the test dataset. The ROC curve performs an excellent job for the best model, and this model has a high sensitivity, as shown in Table 3. To put it another way, the model is highly accurate, with an accuracy of 0.996489 when tested across validation and test datasets.

AUC values for the four basic models and the ensemble model using the IoT Platform dataset are shown in the ROC graph above. We can observe that the stacked ensemble model has the highest accuracy based on the AUC value. After analyzing the AUC values of various models, the stacked ensemble model emerged as the most effective model for predicting mqtt attacks. We can retrieve the AUC value and ROC curve of the best model for the IoT platform test dataset, as shown in Figure 4. The ROC curve does an excellent job for the best models, with high precision and specificity.

Data is provided in **Error! Reference source not found.** After tests were conducted using five machine learning algorithms with two dataset sources acquired from research [15]. While Table 3 shows the datasets obtained from the Indonesian oil service company's system on the IoT platform. In a study [15], the ensemble method outperformed the linear method on three different data sets; The best-achieved accuracy for DoS is 0.99377 using the random forest model, Boosting Gradient reaches 0.99373, and SVM reaches 0.99023 when the two best and worst models are combined. Because we used hyperparameter to tweak the model, we improved the accuracy of the Boosting Gradient and the random forest model in this study, as shown in Table 2. Furthermore, when the random forest was the best model in the previous study with an accuracy of 0.99377, we improved by utilizing a stacked ensemble as the best model with an accuracy of 1.0000.

Table 2 : Results of evaluation metrics

Model	Data Cleansing	Precision	Specificity	Accuracy	Previous Work Accuracy	F-Beta Score
GLM	Include all data	0.998868	0.999862	0.995709		0.995454
GLM	Remove Novariance	0.997742	0.999963	0.9967196		0.988904
Random Forest	Include all data	1.000000	1.000000	0.999985	0.99377	0.999985
Random Forest	Remove Novariance	0.999907	0.999927	0.9995947		0.997152
GBM	Include all data	1.000000	1.000000	0.999985	0.99373	0.999985
GBM	Remove Novariance	1.000000	1.000000	0.9996516		0.996143
Deeplearning	Include all data	1.000000	1.000000	0.999343	0.960836	0.999304
Deeplearning	Remove Novariance	1.000000	1.000000	0.9965427		0.990002
Stacked Ensemble	Include all data	1.000000	1.000000	1.000000		1.000000
Stacked Ensemble	Remove Novariance	1.000000	1.000000	0.9996565		0.997011

Table 3: Dataset from the IoT Platform

Model	Data Cleansing	Precision	Specificity	Accuracy	F-Beta Score
GLM	Include all data	0.994624	0.999715	0.855473	0.820984
Random Forest	Include all data	1.000000	1.000000	0.996424	0.996271
GBM	Include all data	1.000000	1.000000	0.995999	0.995705
Deeplearning	Include all data	1.000000	1.000000	0.991695	0.991104
Stacked Ensemble	Include all data	1.000000	1.000000	0.996489	0.996271

Ensemble technique is the machine learning algorithm with the best accuracy in this study, improving accuracy from previous studies [15]. There are also advances in accuracy on several machine learning. Meanwhile, the stacked ensemble is the machine learning algorithm with the best accuracy for the dataset generated by the IoT platform. The data collected from the system used by the Indonesian oil service company daily is combined with data on attacks on the same system in the dataset obtained from the IoT platform.

Comparing the various datasets and the differences in the preprocessing data indicate high accuracy and F1 scores for the dataset that incorporates all data due to a large number of entries in the dataset. Finally, the results achieved for the discussed machine learning techniques are evaluated based on the accuracy and F1 scores, demonstrating how the model may be used for detection systems on existing systems.

In total, five approaches were tested and evaluated: Logistic Regression, Random Forest, Gradient Boosting Machine, Neural Network, and Stacking Ensemble Model. On the other side, mqtt.topic, eth.dst, mqtt.msg, ip.dst, and eth.src are more critical in determining if a message is an attack or a normal communication. Models that used the Stacked Ensemble method and the Gradient Boosting Machine were also more precise than others. Stacked ensembles produce good results because they can combine the capabilities of several high-performing models to construct predictions that outperform any single model in the ensemble on a classification or regression objective. Overall, all tested models were entirely accurate when making predictions about classified communication.

These results validate the stacked ensemble's accuracy, precision, and specificity on the IoT platform dataset. We improved the [15] dataset using hyperparameters on the H2O engine and all available data instead of limiting the dataset, which yielded better results.

The model must be implemented on the IoT system platform to detect mqtt attacks. Two mqtt brokers with distinct functions can be used to implement the Framework. The first broker receives all communications without filtering them, whereas the second is for IoT platform systems. A scalable service between the two brokers will receive and filter messages from the mirroring broker. The service will classify incoming messages based on the findings of machine learning models. The service will store the clientid in a database if the incoming message is classified as an attack. Still, it will transfer to the IoT platform's broker if the incoming message is classified as a normal message. When messages classed as attacks are received, the database will be updated. As shown in Figure 5, the mirrored broker uses the database to filter the clientid of each incoming message.

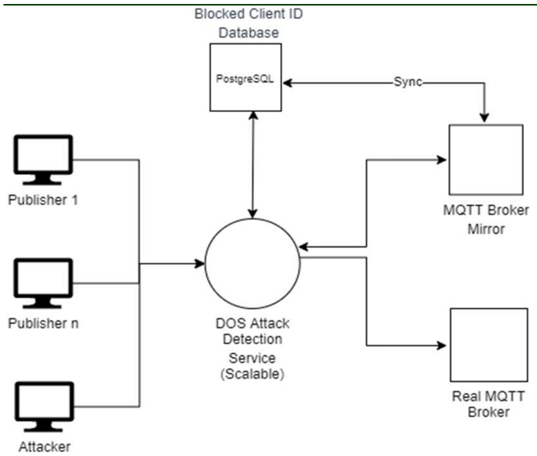

Figure 5: Proposed Framework

The proposed solution is tested by deploying it on an IoT platform. The model utilized is an H2O-based stacked ensemble model.

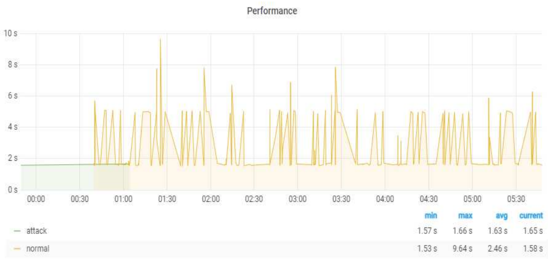

Figure 6: H2O-based stacked ensemble model performance

As seen in Figure 6, the stacked ensemble model's API performance is lower. In an average of 2.46 seconds, the model can respond to the type of data classification (attack or normal) in the input data, with the fastest time being 1.53 seconds. When transferring more data on the IoT platform, this performance will result in long lines. The queue will slow down the model's performance, as seen by the model's longest response time of 9.64 seconds.

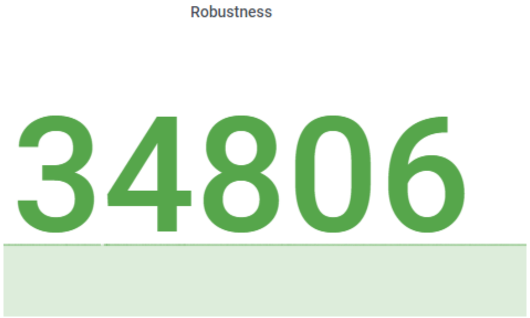

Figure 7: The total amount of data that has been successfully processed.

However, with different results, the model's resilience is shown in Figure 7, that the model can process data in the amount of 34806 for four days. Compared with the incoming data, which is 35502, the amount of data used is not as much as 696, while the data serving is 98.04%. This result shows that the implemented model can meet the reliability aspect

5. CONCLUSION

Systems of IoT have advanced in recent times and are forecasted to increase significantly in the future. Network technologies are more heterogeneous than traditional networks due to the specific characteristics of these devices, posing new cybersecurity issues. We present a methodology for identifying and preventing DoS attacks in IoT systems, considering that attack detection is a critical security barrier that can swiftly discover network attacks and security vulnerabilities.

We leverage datasets from previous studies, simulations, and real-world IoT systems. In this circumstance, frames that the proposed Framework can categorize as attacks or regular messages can be classified using machine learning algorithms. Because binary classification uses a model that predicts the Bernoulli probability distribution for each occurrence, we used binary and multiclass approaches for this classification problem.

These five classification methods are highly effective and can be used in real-world applications. The ensemble technique produced the best results, and the random forest model had better outcomes in general but not as well as the ensemble method. Using the proposed Framework, these models can avoid DoS attacks on IoT systems.

The proposed Framework's resiliency can be used for near-realtime attack detection. The transformation of incoming data into the H2O data frame will take a few seconds in real-time attack detection.

The model developed in this study can be utilized as a guide by organizations working in the IoT industry to increase security on the system/mqtt platform's broker. Companies or organizations should focus on time-based variables in the future to detect DoS attacks on the MQTT protocol. This research shows that time-based factors are vital in identifying MQTT protocol attacks. Furthermore, compared to earlier studies, the usage of hyperparameters in determining the most optimum variable values has been shown to produce improved accuracy, specificity, and precision results [15]. This study shows that adding numerous base learners does not necessarily improve accuracy in the stacked ensemble model.

The performance of the deployment model constructed utilizing the H2O platform is not particularly good at managing real-time data, as can be observed from the performance of the deployment model. In the future study, a model is highly recommended to be deployed on the H2O platform utilizing MOJO (Model Object, Optimized). MOJO enables the model to detect MQTT protocol attacks in real-time.

REFERENCES:

[1] M. T. Okano, “IOT and Industry 4.0: The Industrial New Revolution,” *ICMIS-17 - Int. Conf. Manag. Inf. Syst.*, no. September, pp. 75–82, 2017.

[2] L. Da Xu, E. L. Xu, and L. Li, “Industry 4.0: State of the art and future trends,” *Int. J. Prod. Res.*, vol. 56, no. 8, pp. 2941–2962, 2018, doi: 10.1080/00207543.2018.1444806.

[3] K. Mansoor, A. Ghani, S. A. Chaudhry, S. Shamshirband, S. A. K. Ghayyur, and A. Mosavi, “Securing IoT-based RFID systems: A robust authentication protocol using symmetric cryptography,” *Sensors (Switzerland)*, vol. 19, no. 21, pp. 1–21, 2019, doi: 10.3390/s19214752.

[4] W. D. Hoyer, M. Kroschke, B. Schmitt, K. Kraume, and V. Shankar, “Transforming the Customer Experience Through New Technologies,” *J. Interact. Mark.*, vol. 51, pp. 57–71, 2020, doi: 10.1016/j.intmar.2020.04.001.

[5] R. M. Haris and S. Al-Maadeed, “Integrating Blockchain Technology in 5G enabled IoT: A Review,” *2020 IEEE Int. Conf. Informatics, IoT, Enabling Technol. ICIoT 2020*, pp. 367–371, 2020, doi: 10.1109/ICIoT48696.2020.9089600.

[6] M. Imdad, D. W. Jacob, H. Mahdin, Z. Baharum, S. M. Shaharudin, and M. S. Azmi, “Internet of things: security requirements, attacks and counter measures,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 3, pp. 1520–1530, Jun. 2020, Accessed: Sep. 11, 2021. [Online]. Available: http://ijeecs.iaescore.com/index.php/IJEECS/article/view/21545.

[7] “5.8 Bn Enterprise & Automotive IoT Endpoints to be Used | Gartner.” https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io (accessed Jul. 16, 2021).

[8] T. Prasetya, N. Harsono, and R. A. Pratama, *Buku Putih Indonesia ICT Industry*. 2021.

[9] G. ur Rehman, A. Ghani, M. Zubair, S. A. K. Ghayyure, and S. Muhammad, “Honesty based democratic scheme to improve community cooperation for Internet of Things based vehicular delay tolerant networks,” *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, pp. 1–19, 2021, doi: 10.1002/ett.4191.

[10] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, “Authorization mechanism for MQTT-based Internet of Things,” 2016, doi: 10.1109/ICCW.2016.7503802.

[11] D. H. Mun, M. Le Dinh, and Y. W. Kwon, “An Assessment of Internet of Things Protocols for Resource-Constrained Applications,” 2016, doi: 10.1109/COMPSAC.2016.51.

[12] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, and G. Reali, “Comparison of two lightweight protocols for smartphone-based sensing,” 2013, doi: 10.1109/SCVT.2013.6735994.

[13] N. Naik, “Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP,” 2017, doi: 10.1109/SysEng.2017.8088251.

[14] S. Andy, B. Rahardjo, and B. Hanindhito, “Attack scenarios and security analysis of mqtt communication protocol in iot system,” 2017, doi: 10.11591/eecsi.4.1064.

[15] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, and C. Benavides, “Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol,” *Complexity*, 2019, doi: 10.1155/2019/6516253.

[16] A. M. Abdul and S. Umar, “Attacks of denial-of-service on networks layer of OSI model and maintaining of security,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 5, no. 1, pp. 181–186, 2017, doi: 10.11591/ijeecs.v5.i1.pp181-186.

[17] E. Ciklabakkal, A. Donmez, M. Erdemir, E. Suren, M. K. Yilmaz, and P. Angin, “ARTEMIS: An intrusion detection system for mqtt attacks in internet of things,” *Proc. IEEE Symp. Reliab. Distrib. Syst.*, pp. 369–371, 2019, doi: 10.1109/SRDS47363.2019.00053.

[18] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," *IEEE Access*. 2019, doi: 10.1109/ACCESS.2019.2924045.

[19] G. Potrino, F. De Rango, and A. F. Santamaria, "Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker," 2019, doi: 10.1109/WCNC.2019.8885553.

[20] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine learning based IoT intrusion detection system: An MQTT case study," *arXiv*. 2020.

[21] N. F. Syed, Z. Baig, A. Ibrahim, and C. Valli, "Denial of service attack detection through machine learning for the IoT," *J. Inf. Telecommun.*, vol. 4, no. 4, 2020, doi: 10.1080/24751839.2020.1767484.

[22] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "Mqttset, a new dataset for machine learning techniques on mqtt," *Sensors (Switzerland)*, vol. 20, no. 22, 2020, doi: 10.3390/s20226578.

[23] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, vol. 2017-Decem, no. May, 2017, doi: 10.1109/EECSI.2017.8239179.

[24] H. Hejazi, H. Rajab, T. Cinkler, and L. Lengyel, "Survey of platforms for massive IoT," *2018 IEEE Int. Conf. Futur. IoT Technol. Futur. IoT 2018*, vol. 2018-Janua, pp. 1–8, 2018, doi: 10.1109/FIOT.2018.8325598.

[25] M. Z. Imtiyaz, M. Nasrun, and U. A. Ahmad, "Analisis Dan Implementasi Framework Crisp-Dm Untuk Mengetahui Perilaku Data Transaksi Pelanggan," *e-Proceeding Eng.*, 2015.

[26] F. Martinez-Plumed *et al.*, "CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 8, pp. 3048–3061, 2021, doi: 10.1109/TKDE.2019.2962680.

[27] A. Sanchez, R. Basanya, T. Janowski, and A. Ojo, "Enterprise Architectures – Enabling Interoperability Between Organizations," *Framework*, 2007.

[28] "etactica/mqtt-malaria: Attacking MQTT systems with Mosquittos (scalability and load testing utilities for MQTT environments)." https://github.com/etactica/mqtt-malaria (accessed Jul. 17, 2021).

[29] D. Suleiman and G. Al-Naymat, "ScienceDirect SMS Spam Detection using H2O Framework," *Procedia Comput. Sci.*, vol. 113, pp. 154–161, 2017, doi: 10.1016/j.procs.2017.08.335.

[30] R. Pari, M. Sandhya, and S. Sankar, "A Multitier Stacked Ensemble Algorithm for Improving Classification Accuracy," *Comput. Sci. Eng.*, vol. 22, no. 4, pp. 74–85, 2020, doi: 10.1109/MCSE.2018.2873940.

[31] H. M and S. M.N, "A Review on Evaluation Metrics for Data Classification Evaluations," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 01–11, 2015, doi: 10.5121/ijdkp.2015.5201.

[32] M. Shafiq, X. Yu, A. K. Bashir, H. N. Chaudhry, and D. Wang, "A machine learning approach for feature selection traffic classification using security analysis," *J. Supercomput.*, vol. 74, no. 10, pp. 4867–4892, 2018, doi: 10.1007/s11227-018-2263-3.