# Machine Learning Model for Message Queuing Telemetry Transport Data Analytics

**B. HARITHA LAKSHMI[1], M. NAVYASRI[2], M. PRASANNA[2], M. MANASA[2]**

[1]Assistant Professor, Department of Information Technology, Mallareddy Engineering College for Women, (UGC-Autonomous), Hyderabad, India, harithamrecw@gmail.com.

[2]Student, Department of Information Technology, Mallareddy Engineering College for Women, (UGC-Autonomous), Hyderabad, India.

## Abstract

In the realm of MQTT data analytics, this project aims to accomplish the primary objective of developing and evaluating machine learning models for the classification of MQTT messages into distinct categories or message types. Two prominent classifiers, the Random Forest Classifier and K-Nearest Neighbors (KNN) Classifier, have been effectively implemented and rigorously assessed for their classification capabilities. The assessment primarily involves measuring the accuracy and precision of both models, employing comprehensive classification reports to ascertain their competence in categorizing MQTT data. The results of this evaluation reveal that both the Random Forest and KNN classifiers excel in effectively classifying MQTT messages with remarkable accuracy and precision. A standout feature of this project is the incorporation of confusion matrices to visually represent the performance of these models. These matrices offer a clear and intuitive depiction of the models' effectiveness by showcasing key metrics such as true positives, true negatives, false positives, and false negatives. This visual representation provides invaluable insights into the strengths and weaknesses of the classification models, aiding in a deeper understanding of their performance. In summary, this project sheds light on the potential of machine learning models, specifically Random Forest and KNN classifiers, in the context of MQTT data analytics. It highlights their proficiency in accurately categorizing MQTT messages, further enhanced by the visual clarity offered by confusion matrices. The findings presented in this study serve as a foundation for future research and applications in the domain of MQTT data analysis and classification.

## 1. Introduction

When the Internet of Things (IoT) is implemented, physical devices (also known as IoT nodes) are connected to the internet, enabling them to collect and exchange data with other nodes in the network without the need for human participation [1]. Message queuing telemetry transfer (MQTT) gained widespread use in a range of applications, such as in smart homes [2–4], agricultural IoT [5, 6], and industrial applications. This is mainly due to its capacity to communicate at low bandwidths, the necessity for minimum memory, and reduced packet loss. Figure 1 depicts the architecture of the MQTT protocol for use in the IoT. The IoT and associated technologies evolved at a rapid rate, with 15 billion linked devices in 2015, which is likely to increase to 38 billion devices by 2025, according to Gartner [7]. The IoT is a network of objects—linked by sensors, actuators, gateways, and cloud services—that delivers a service to users. The MQTT protocol was integrated into a number of IoT applications. Figure 2 depicts how the MQTT protocol maintains IoT applications. Traditional intrusion detection systems (IDSs) are only successful when dealing with data that move slowly or with small volumes of data. They are currently inefficient when dealing with big data or networks and are unable to cope with high-speed data transmission. Therefore, technologies capable of dealing with massive volumes of data and identifying any indications of network penetration are crucial. When it comes to big data, data security and privacy are perhaps the most pressing concerns, especially in the context of network assaults. Distributed denial-of-service (DDoS) attacks are one of the most common types of cyberattacks. They target servers or networks with the intent of interfering with their normal operation. Although real-time detection and mitigation of DDoS attacks is difficult to achieve, a solution would be extremely valuable, since attacks can cause significant damage.

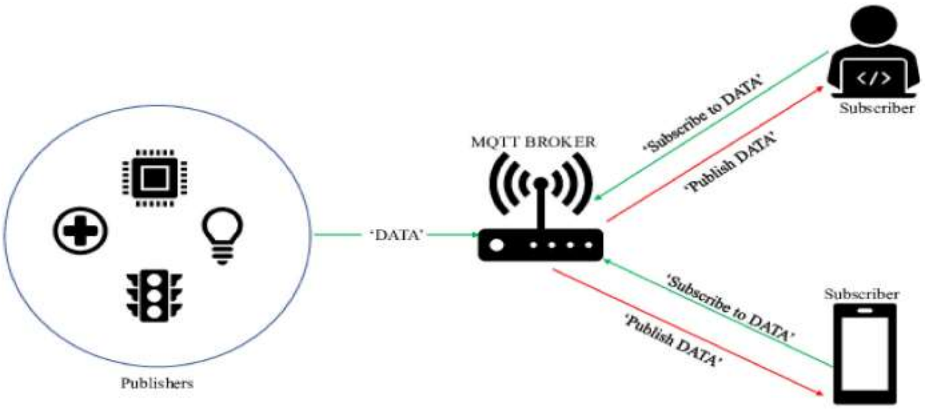


Figure 1: Topology of the MQTT protocol.

## 2. Literature Survey

Machine learning (ML) studies are always being improved through the use of training data and the exploitation of available information. Some consider ML to be a component of artificial intelligence. Depending on the information provided, various types of learning can be undertaken, including supervised learning—for example, the support vector machine (SVM) algorithm and the k-nearest neighbors (KNN) algorithm—semi-supervised learning, and unsupervised learning (e.g., clustering methods). Deep learning (DL) models combined with ML techniques produce excellent results in cybersecurity systems used for detecting attacks. ML techniques are used in multiple contexts, such as in healthcare. For example, they are being used to forecast COVID-19 outbreaks, osteoporosis, and schistosomiasis, among other health-related problems [8, 9]. Many researchers employed classification algorithms to detect and resolve DDoS attacks with the goal of reducing the number of attacks. DDoS attacks are simple to carry out because they take advantage of network flaws and generate requests for software services [10, 11]. DDoS attacks take a long time to identify and neutralize, and this solution is particularly useful, since these attacks may cause major harm. There are significant drawbacks to the current methods used to detect DDoS attacks, such as high processing costs and the inability to handle enormous quantities of data reaching the server [12]. Using a variety of classification methods, classification algorithms differentiate DDoS packets from other kinds of packets [13–15]. To secure the IoT against anomalous adversarial attacks, various security-enhancing solutions were developed. These approaches are often used to detect attacks in IoT networks by monitoring IoT node operations, such as the rate at which data are sent. In this paper, we introduce a brief review of the literature to highlight recent advancements in IoT security systems, with a particular emphasis on IDSs that target the MQTT protocol. The authors of [16] provided a process tree-based intrusion detection technique for MQTT protocols based on their previous work. It describes network behavior in terms of the hierarchical branches of a tree, which can then be used to detect assaults or aberrant behavior in the network. The detection rate was used to evaluate the model, and four frequent types of assaults were introduced into the network to assess its performance. However, little consideration was given to newly created adversarial attacks and intrusions.

The study [17] proposes a fuzzy logic model for intrusion detection that is specifically built to safeguard IoT nodes that use the MQTT protocol from denial-of-service (DoS) attacks. Although fuzzy logic demonstrated its efficacy in a variety of systems, including sensor device intrusion detection in the IoT [18], its high difficulty with increasing input dimensions limits its ability to detect attacks on IoT platforms where large amounts of data are transferred on a continuous basis. The extreme gradient boosting (XGBoost) algorithm gated recurrent units (GRUs), and LSTM are only a few of the ML methods used in [19, 20] to create a cybersecurity system for the MQTT protocol in the IoT. The MQTT dataset, which contains three forms of attacks, including intrusion (illegal entrance), DoS, and malicious code injection and man-in-the-middle attack (MitM), was used to verify the proposed techniques. To test a range of ML approaches, the MQTT-IoTIDS2020 dataset was used. Using these ML approaches, it was found that a system for detecting MQTT attacks could be designed, and this was later validated by the researchers. An MQTT-enabled IoT cybersecurity system demonstrated the use of an ANN approach for intrusion detection [21].

Ujjan et al. [22] presented an entropy-based features section to identify the important features in network traffic for detecting DoS attacks by employing an encoder (SAE) and CNN models. CPU consumption was significantly higher and took significantly longer. The models were accurate to within 94% and 93% of the true value, respectively. Using LSTM and CNN, Gadze et al. [23] introduced DL models to identify DDoS intrusions on a software-defined network's centralized controller (CNN). The accuracy of the models was lower than expected. When data were split out in a 70/30 ratio, the accuracies of LSTM and CNN were 89.63% and 66%, respectively. However, when using an LSTM model to detect intrusions in network traffic, DDoS was found to be the most time-consuming attempt out of all 10 attempts tested. A hybrid ML model, SVM combined with random forest (SVC-RF), was created by Ahuja et al. [24] and used to distinguish between benign and malicious traffic. The authors extracted features from the original dataset that were used to build a new dataset: the SDN dataset, which had innovative features. It was determined that the SVC-RF classifier is capable of accurately categorizing data traffic with an accuracy of 98.8% when using the software defined networking (SDN) dataset. Wang et al. [25] revealed that a unique DL model based on an upgraded deep belief network (DBN) can be used to identify network intrusions more quickly. They replaced the back propagation approach in DBN with a kernel-based extreme learning machine (KELM), which was created by the researchers and is still in development. Their model outperformed other current neural network approaches by a wide margin. In this study, the researchers examined and tested the accuracy of a number of different categorization algorithms and techniques. The results reveal that the DBN-KELM algorithm obtained an accuracy of 93.5%, while the DBN-EGWO-KELM method achieved an accuracy of 98.60%.

## 3. Proposed System Design

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

### Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

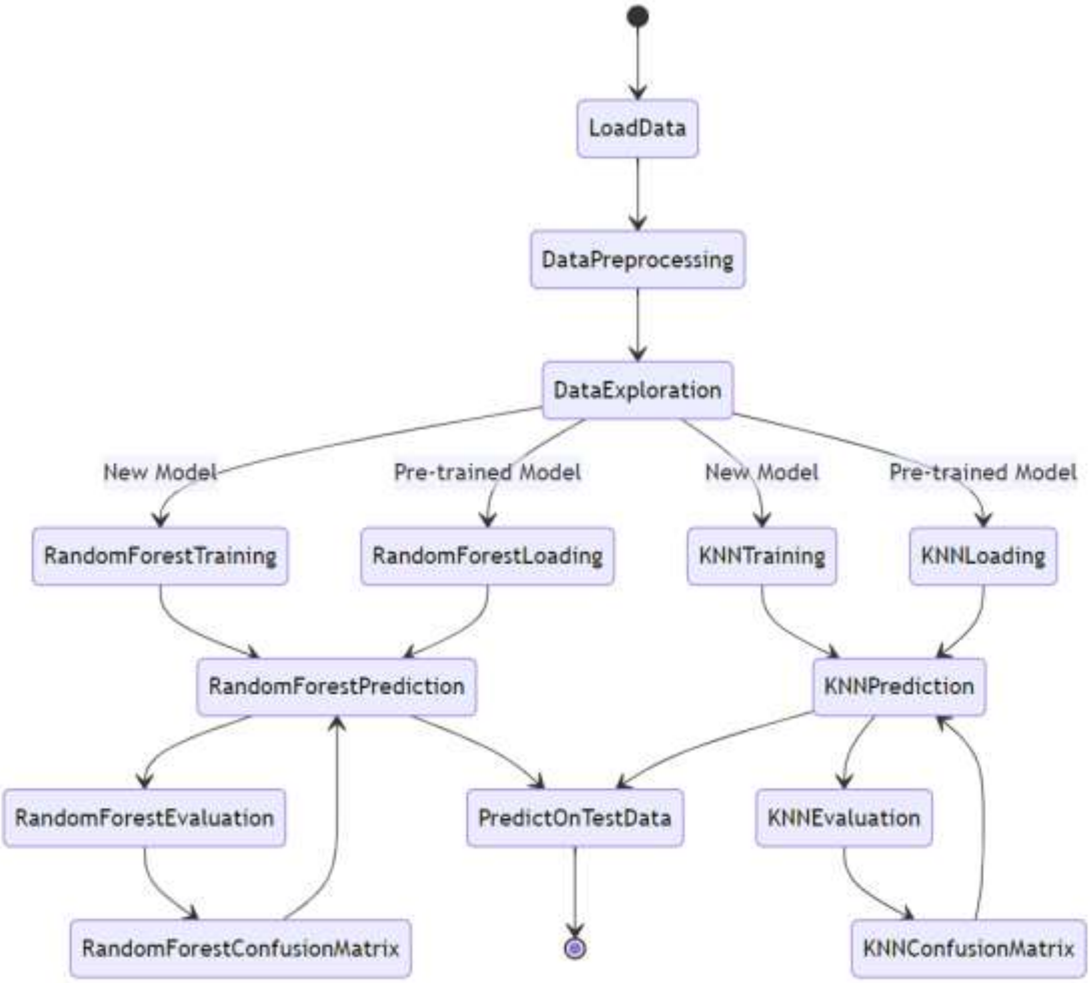


Figure 1. Proposed system model.

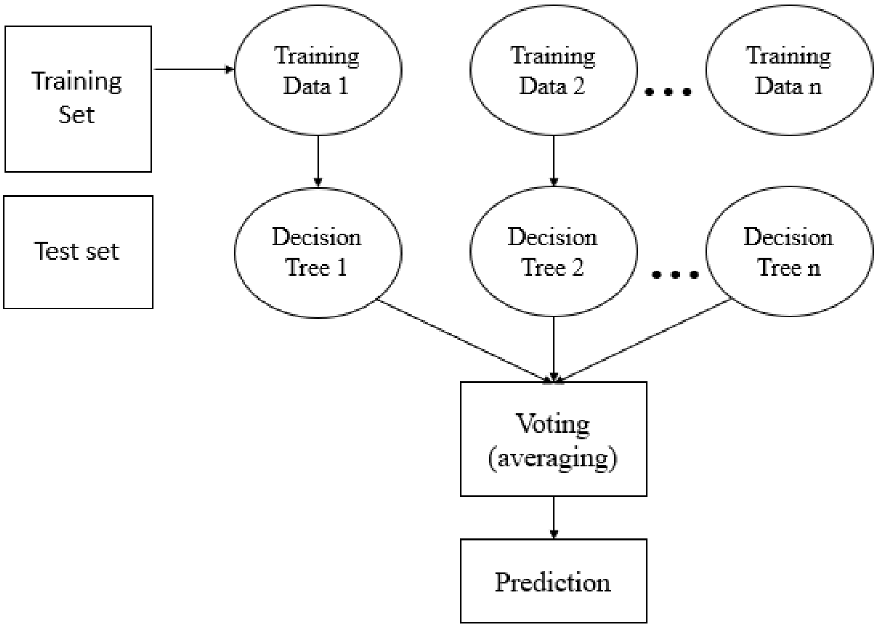


Figure 2: Random Forest algorithm.

**Random Forest algorithm**

Step 1: In Random Forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

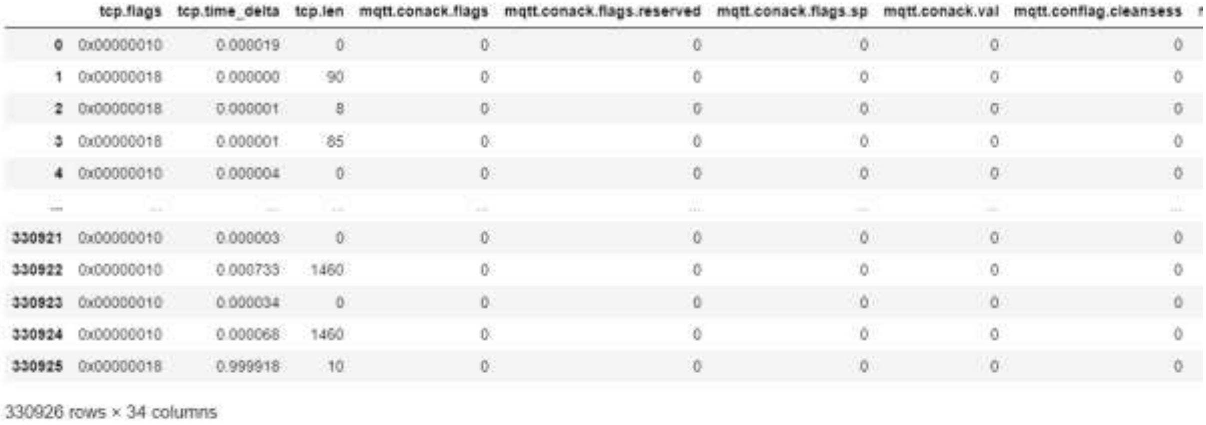
Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

**Important Features of Random Forest**

- **Diversity-** Not all attributes/variables/features are considered while making an individual tree, each tree is different.
- **Immune to the curse of dimensionality-** Since each tree does not consider all the features, the feature space is reduced.
- **Parallelization**-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.
- **Train-Test split-** In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.
- **Stability-** Stability arises because the result is based on majority voting/ averaging.

## 4. Results and description

Figure 3 represents a graphical representation of the MQTT dataset used in the project. It shows a portion of the dataset with rows and columns, where each row represents a data point (e.g., an MQTT message) and columns represent different features (e.g., message attributes).



| | tcp.flags | tcp.time_delta | tcp.len | mqtt.conack.flags | mqtt.conack.flags.reserved | mqtt.conack.flags.sp | mqtt.conack.val | mqtt.conflag.cleansess | r |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0x00000010 | 0.000019 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **1** | 0x00000018 | 0.000000 | 90 | 0 | 0 | 0 | 0 | 0 | |
| **2** | 0x00000018 | 0.000001 | 8 | 0 | 0 | 0 | 0 | 0 | |
| **3** | 0x00000018 | 0.000001 | 85 | 0 | 0 | 0 | 0 | 0 | |
| **4** | 0x00000010 | 0.000004 | 0 | 0 | 0 | 0 | 0 | 0 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **330921** | 0x00000010 | 0.000003 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **330922** | 0x00000010 | 0.000733 | 1460 | 0 | 0 | 0 | 0 | 0 | |
| **330923** | 0x00000010 | 0.000034 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **330924** | 0x00000010 | 0.000068 | 1460 | 0 | 0 | 0 | 0 | 0 | |
| **330925** | 0x00000018 | 0.999918 | 10 | 0 | 0 | 0 | 0 | 0 | |

330926 rows × 34 columns

Figure 3: Sample dataset used for MQTT data analytics.

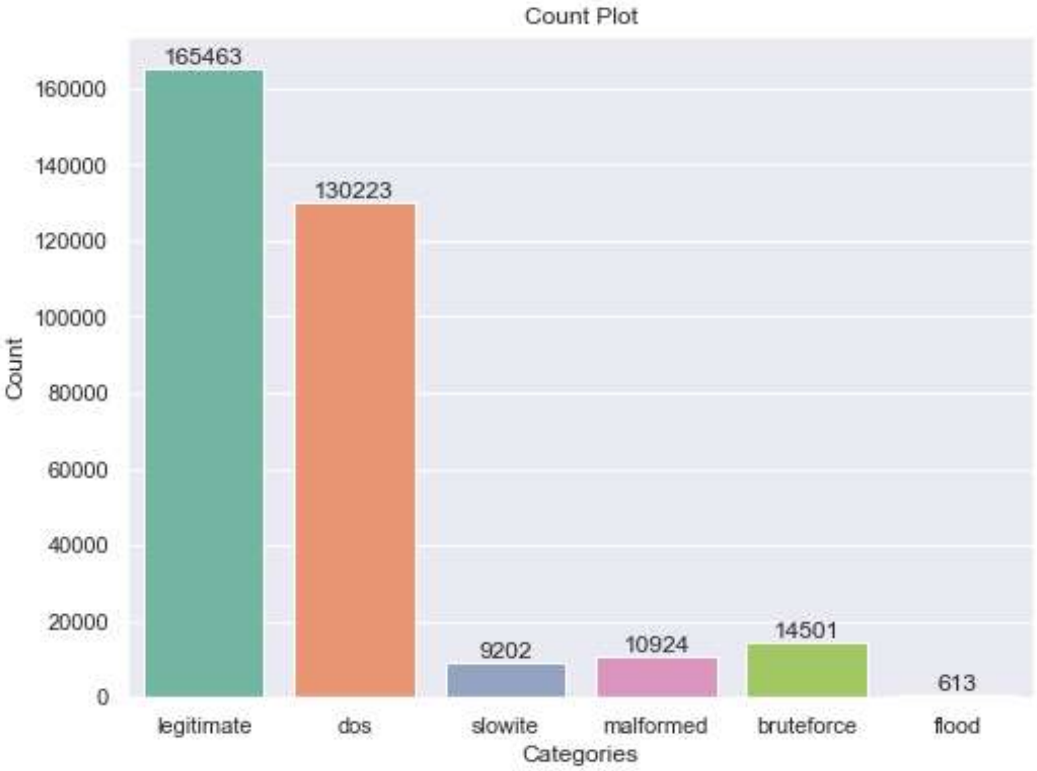
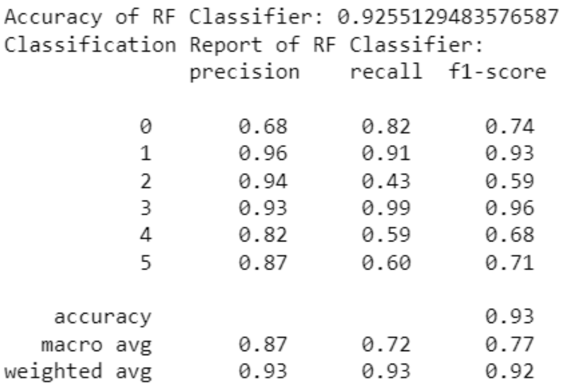


Figure 4: Count plot of target categories.



```
Accuracy of RF Classifier: 0.9255129483576587
Classification Report of RF Classifier:
              precision    recall  f1-score

           0       0.68      0.82      0.74
           1       0.96      0.91      0.93
           2       0.94      0.43      0.59
           3       0.93      0.99      0.96
           4       0.82      0.59      0.68
           5       0.87      0.60      0.71

    accuracy                           0.93
   macro avg       0.87      0.72      0.77
weighted avg       0.93      0.93      0.92
```

Figure 5: Classification report and accuracy obtained using RF classification.

## 5. Conclusion

In the context of MQTT data analytics, this project has successfully accomplished its primary objective of developing and evaluating machine learning models for classifying MQTT messages into distinct categories or message types. The Random Forest Classifier and KNN Classifier have been effectively implemented and assessed for their classification capabilities. Through accuracy measurements and the generation of comprehensive classification reports, both models have demonstrated their competence in categorizing MQTT data with accuracy and precision. One notable feature of the project is the visualization of model performance using confusion matrices. These matrices provide a clear representation of the models' effectiveness by depicting true positives, true negatives, false positives, and false negatives. Such visual insights are invaluable for understanding the models' strengths and weaknesses.

## References

[1] Al-Masri, E.; Kalyanam, K.R.; Batts, J.; Kim, J.; Singh, S.; Vo, T.; Yan, C. Investigating messaging protocols for the Internet of Things (IoT). IEEE Access 2020, 8, 94880–94911.

[2] Kodali, R.K.; Soratkal, S. MQTT Based Home Automation System Using ESP8266. In Proceedings of the 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Agra, India, 21–23 December 2016; pp. 1–5.

[3] Cornel-Cristian, A.; Gabriel, T.; Arhip-Calin, M.; Zamfirescu, A. Smart Home Automation with MQTT. In Proceedings of the 2019 54th International Universities Power Engineering Conference (UPEC), Bucharest, Romania, 3–6 September 2019; pp. 1–5.

[4] Prabaharan, J.; Swamy, A.; Sharma, A.; Bharath, K.N.; Mundra, P.R.; Mohammed, K.J. Wireless Home Automation and Securitysystem Using MQTT Protocol. In Proceedings of the 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 19–20 May 2017; pp. 2043–2045.

[5] Kodali, R.K.; Sarjerao, B.S. A Low Cost Smart Irrigation System Using MQTT Protocol. In Proceedings of the 2017 IEEE Region 10 Symposium (TENSYMP), Cochin, India, 14–16 July 2017; pp. 1–5.

[6] Mukherji, S.V.; Sinha, R.; Basak, S.; Kar, S.P. Smart Agriculture Using Internet of Things and mqtt Protocol. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 14–16.

[7] Rayes, A.; Salam, S. Internet of Things from Hype to Reality—The Road to Digitization, 2nd ed.; Springer: Cham, Switzerland, 2019.

[8] Anam, M.; Ponnusamy, V.; Hussain, M.; Nadeem, M.W.; Javed, M.; Goh, H.G.; Qadeer, S. Osteoporosis Prediction for Trabecular Bone Using Machine Learning: A Review. Comput. Mater. Contin. 2021, 67, 89–105.

[9] Polat, H.; Polat, O.; Cetin, A. Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models. Sustainability 2020, 12, 1035.

[10] Ochôa, I.S.; Leithardt, V.R.Q.; Calbusch, L.; Santana, J.F.D.P.; Parreira, W.D.; Seman, L.O.; Zeferino, C.A. Performance and Security Evaluation on a Blockchain Architecture for License Plate Recognition Systems. Appl. Sci. 2021, 11, 1255.

[11] Anjos, J.C.S.D.; Gross, J.L.G.; Matteussi, K.J.; González, G.V.; Leithardt, V.R.Q.; Geyer, C.F.R. An Algorithm to Minimize Energy Consumption and Elapsed Time for IoT Workloads in a Hybrid Architecture. Sensors 2021, 21, 2914.

[12] Ganguly, S.; Garofalakis, M.; Rastogi, R.; Sabnani, K. Streaming Algorithms for Robust, Real-Time Detection of ddos Attacks. In Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS'07), Toronto, ON, Canada, 25–27 June 2007; p. 4.

[13] Soni, D.; Makwana, A. A Survey on mqtt: A Protocol of Internet of Things (Iot). In Proceedings of the International Conference on Telecommunication, Power Analysis and Computing Techniques (ICTPACT-2017), Chennai, India, 6–8 April 2017; Volume 20.

[14] Hunkeler, U.; Truong, H.L.; Stanford-Clark, A. MQTT-S—A Publish/Subscribe Protocol for Wireless Sensor Networks. In Proceedings of the 2008 3rd International Conference on

Communication Systems Software and Middleware and Workshops (COMSWARE'08), Bangalore, India, 6–10 January 2008; pp. 791–798.

[15] Ahmadon, M.A.B.; Yamaguchi, N.; Yamaguchi, S. Process-Based Intrusion Detection Method for IoT System with MQTT Protocol. In Proceedings of the 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 15–18 October 2019; pp. 953–956.

[16] Jan, S.U.; Lee, Y.D.; Koo, I.S. A distributed sensor-fault detection and diagnosis framework using machine learning. Inf. Sci. 2021, 547, 777–796.

[17] Alaiz-Moreton, H.; Aveleira-Mata, J.; Ondicol-Garcia, J.; Muñoz-Castañeda, A.L.; García, I.; Benavides, C. Multiclass classification procedure for detecting attacks on MQTT-IoT protocol. Complexity 2019, 2019, 6516253.

[18] Hindy, H.; Bayne, E.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Bellekens, X. Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset). In Proceedings of the International Networking Conference, Online, 19–21 September 2020; Springer: Cham, Switzerland, 2020; pp. 73–84.

[19] Ullah, I.; Ullah, A.; Sajjad, M. Towards a Hybrid Deep Learning Model for Anomalous Activities Detection in Internet of Things Networks. IoT 2021, 2, 428–448.

[20] Almaiah, M.A.; Almomani, O.; Alsaaidah, A.; Al-Otaibi, S.; Bani-Hani, N.; Hwaitat, A.K.A.; Al-Zahrani, A.; Lutfi, A.; Awad, A.B.; Aldhyani, T.H.H. Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels. Electronics 2022, 11, 3571.

[21] Shalaginov, A.; Semeniuta, O.; Alazab, M. MEML: Resource-Aware MQTT-Based Machine Learning for Network Attacks Detection on IoT Edge Devices. In Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion, Auckland, New Zealand, 2–5 December 2019; pp. 123–128.

[22] Ujjan, R.M.A.; Pervez, Z.; Dahal, K.; Khan, W.A.; Khattak, A.M.; Hayat, B. Entropy Based Features Distribution for Anti-DDoS Model in SDN. Sustainability 2021, 13, 1522.

[23] Gadze, J.D.; Bamfo-Asante, A.A.; Agyemang, J.O.; Nunoo-Mensah, H.; Opare, K.A.-B. An Investigation into the Application of Deep Learning in the Detection and Mitigation of DDOS Attack on SDN Controllers. Technologies 2021, 9, 14.

[24] Ahuja, N.; Singal, G.; Mukhopadhyay, D.; Kumar, N. Automated DDOS attack detection in software defined networking. J. Netw. Comput. Appl. 2021, 187, 103108.

[25] Wang, Z.; Zeng, Y.; Liu, Y.; Li, D. Deep belief network integrating improved kernel-based extreme learning machine for network intrusion detection. IEEE Access 2021, 9, 16062–16091.