electronics

Article

Artificial Intelligence Algorithms for Detecting and Classifying MQTT Protocol Internet of Things Attacks

Ali Alzahrani [1] and Theyazn H. H. Aldhyani [2,*]

[1] Department of Computer Engineering, King Faisal University, Al Hofuf 31982, Saudi Arabia
[2] Applied College in Abqaiq, King Faisal University, AI Hofuf 31982, Saudi Arabia
* Correspondence: taldhyani@kfu.edu.sa

Abstract: The Internet of Things (IoT) grew in popularity in recent years, becoming a crucial component of industrial, residential, and telecommunication applications, among others. This innovative idea promotes communication between physical components, such as sensors and actuators, to improve process flexibility and efficiency. Smart gadgets in IoT contexts interact using various message protocols. Message queuing telemetry transfer (MQTT) is a protocol that is used extensively in the IoT context to deliver sensor or event data. The aim of the proposed system is to create an intrusion detection system based on an artificial intelligence algorithm, which is becoming essential in the defense of the IoT networks against cybersecurity threats. This study proposes using a k-nearest neighbors (KNN) algorithm, linear discriminant analysis (LDA), a convolutional neural network (CNN), and a convolutional long short-term memory neural network (CNN-LSTM) to identify MQTT protocol IoT intrusions. A cybersecurity system based on artificial intelligence algorithms was examined and evaluated using a standard dataset retrieved from the Kaggle repository. The dataset was injected by five attacks, namely brute-force, flooding, malformed packet, SlowITe, and normal packets. The deep learning algorithm achieved high performance compared with the developing security system using machine learning algorithms. The performance accuracy of the KNN method was 80.82%, while the accuracy of the LDA algorithm was 76.60%. The CNN-LSTM model attained a high level of precision (98.94%) and is thus very effective at detecting intrusions in IoT settings.

Keywords: cybersecurity; MQTT; deep learning; artificial intelligence; machine learning

Citation: Alzahrani, A.; Aldhyani, T.H.H. Artificial Intelligence Algorithms for Detecting and Classifying MQTT Protocol Internet of Things Attacks. *Electronics* **2022**, *11*, 3837. https://doi.org/10.3390/electronics11223837

Academic Editor: Rameez Asif

Received: 16 September 2022
Accepted: 16 November 2022
Published: 21 November 2022

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 [1,7–9]. Figure 1 depicts the architecture of the MQTT protocol for use in the IoT.

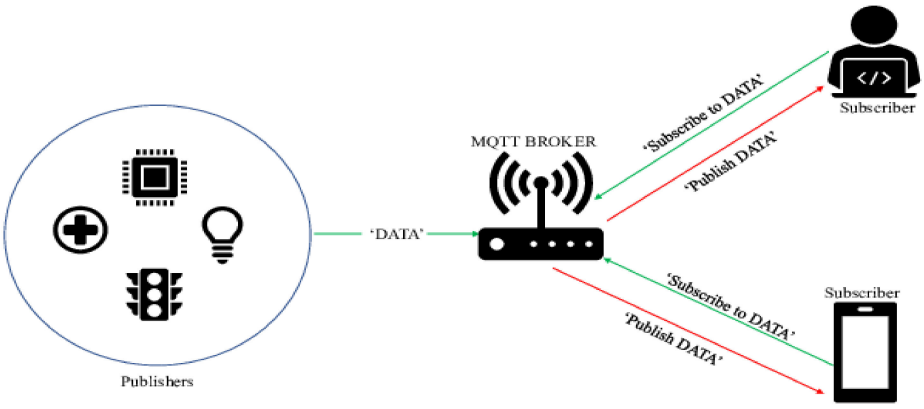

Figure 1. Topology of the MQTT protocol [10].

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 [11]. The IoT is a network of objects—linked by sensors, actuators, gateways, and cloud services—that delivers a service to users [12]. The MQTT protocol was integrated into a number of IoT applications. Figure 2 depicts how the MQTT protocol maintains IoT applications.

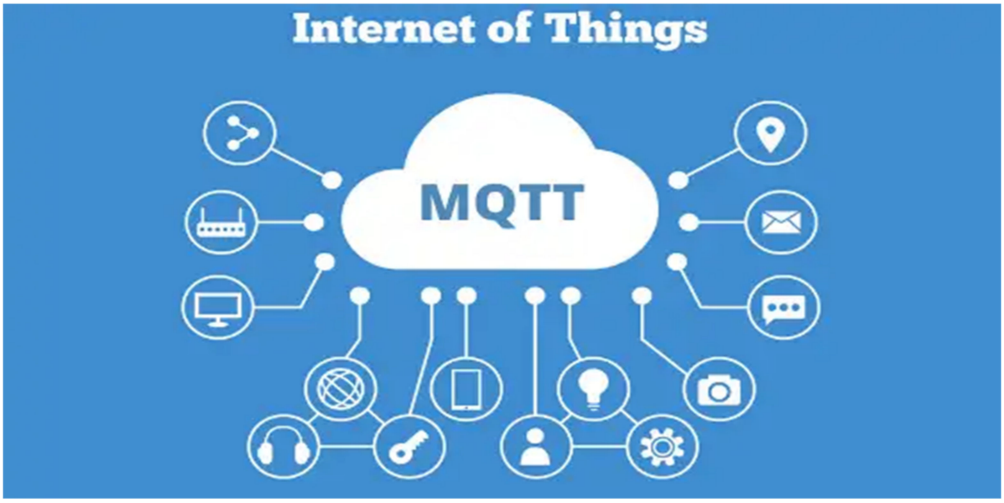

Figure 2. Connecting the MQTT protocol with IoT applications [13].

Traditional intrusion detection systems (IDSs) are only successful when dealing with data that move slowly or with small volumes of data [14,15]. 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 [16]. 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 [17]. 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 [18].

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 [19–24]. 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 [25,26]. 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 [27]. Using a variety of classification methods, classification algorithms differentiate DDoS packets from other kinds of packets [28–30]. 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 [31] 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 [32] 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 [33], 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 [34,35] 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 [36].

Ujjan et al. [37] 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. [38] 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. [39] 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. [40] 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%. The following list summarizes the study's contributions:

- A convolutional long short-term memory neural network (CNN-LSTM) was proposed to detect MQTT protocol IoT intrusions.
- An ML approach to MQTT attack detection, namely the KNN algorithm and linear discriminant analysis (LDA).
- The security system was examined using standard datasets containing MQTT attacks.
- Compared to other current systems, the suggested approach was highly accurate.

2. Materials and Methods

The framework of the MQTT protocol based on the intrusion detection system is presented in Figure 3.

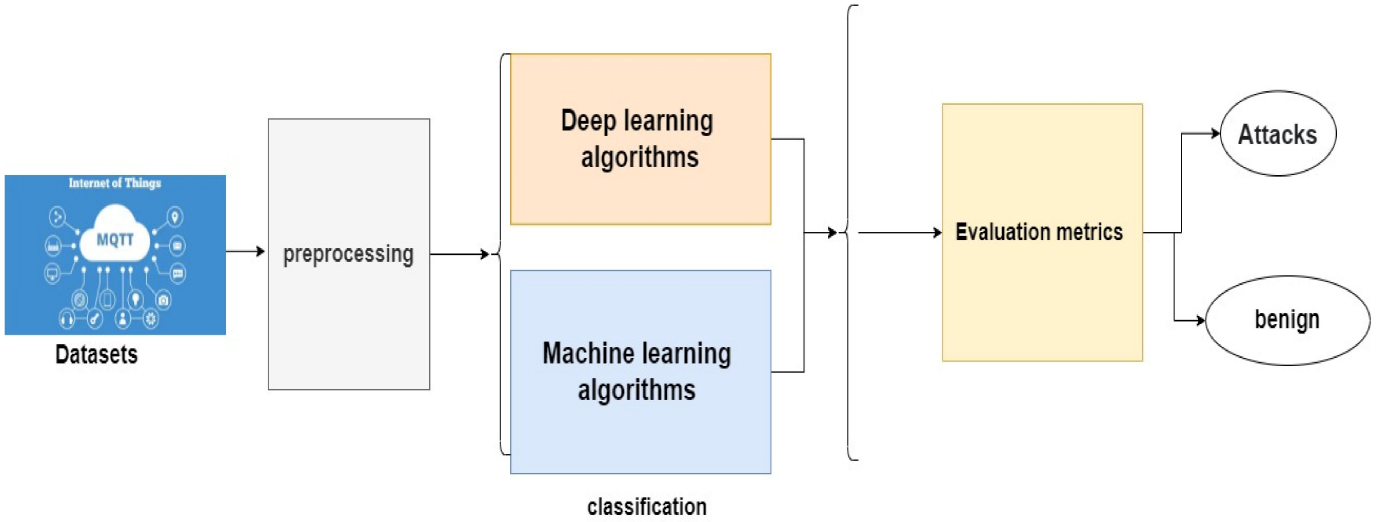

Figure 3. Proposed system for detecting MQTT attacks.

2.1. Datasets

The MQTT protocol is widely used in the IoT to facilitate communication between machines. The proliferation of IoT devices and protocols necessitated the development of new and effective IDSs. However, to implement IoT IDSs, data are needed to analyze, train, and test models. The collection is made up of MQTT-based IoT sensors that represent real-world networks. The MQTT broker was created with Eclipse Mosquitto, and the network had eight sensors. Given that each sensor's behavior differed, the scenario was based on a smart home setting where sensors gather data on temperature, humidity, CO_2 levels, motion, smoke, doors, and fans. Table 1 shows the number of MQTT attacks and the normal. The dataset is available here: https://www.kaggle.com/cnrieiit/mqttset (accessed on 16 July 2022).

The numbers of the classes for the entire MQTT-based intrusion detection system over the IoT is presented in Figure 4.

Table 1. MQTT attacks.

Attacks/Normal	Description
Brute-force	There are a number of ways to carry out brute-force attacks, including the use of trial and error. Every possible combination is tested by hackers with the goal of making a correct guess.
Flooding	Attacks described as flooding are also known as denial-of-service (DoS) attacks. As the name implies, attackers use a flood attack to overwhelm a system with traffic, preventing it from properly inspecting and allowing legitimate network data.
Malformed packet	A malformed packet attack is a kind of single-packet attack. In most single-packet attacks, the attacker employs one of the following techniques: devices malfunction or crash as a result of malicious packets sent by an attacker.
SlowITe	A slow DoS against Internet of Things environments (SlowITe) is a new MQTT DoS attack. It attacks network services while using little bandwidth and resources.

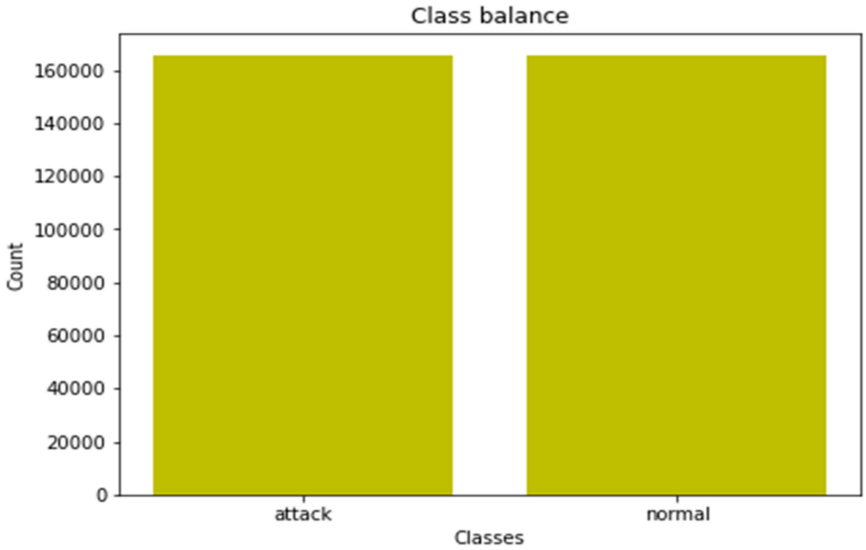

Figure 4. Classes of MQTT datasets.

2.2. Preprocessing Approaches

The IoT has a very complex format because it is connected to different networks; therefore, a preprocessing approach is needed to handle the features of the dataset. For the categorical features, we used a hot encoding function to convert these features into numeric form.

Standard normalization was used to scale the data to the same format, making it easier for the classification algorithm to obtain high accuracy. This was done by subtracting data from the mean and dividing by the standard deviation.

2.3. Proposed Model Based on Classification Algorithms

An explanation of the methods employed in this study, including KNN, LDA, and DL techniques such as CNN or LSTM, is provided in this section.

2.3.1. K-Nearest Neighbors

KNN is a form of learning that requires supervision. The KNN method places the new case or data into a category that is consistent with the instances that are already there by comparing it to the previous cases. The KNN algorithm saves all available data and compares it to previous data points. The KNN method can easily classify fresh data into suitable categories [4–45]. Classification issues are usually solved using the KNN method. In other words, it makes no assumptions about the underlying facts. As a lazy learner

algorithm, KNN saves the training set and then uses it to classify it. This algorithm saves data and subsequently classifies fresh data into comparable categories.

$$E_i = \sqrt{(a_1 - a_2) + (b_1 - b_2)^2}, \quad (1)$$

where a_1, a_2, b_1, and b_2 are the IDS input data.

2.3.2. Linear Discriminant Analysis

Handling high-dimensional applications is a challenge. We can simplify intrusion detection by splitting the data into two categories—normal packets and hazardous packets—and then swapping the data between the two groups using LDA, a linear ML approach [46]. Figure 5 depicts the LDA technique for analyzing intrusion classes and normal classes on the IoT network, with the blue line denoting a linear separation between the two types of data.

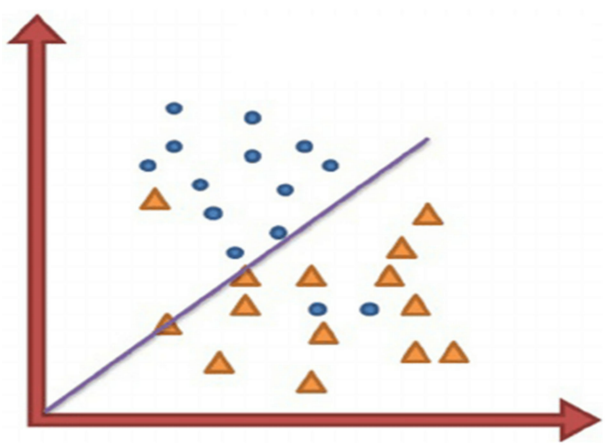

Figure 5. Linear Discriminant Analysis structure for two classes.

2.3.3. Deep Learning Models

CNNs have five layers: an input layer for handling the MQTT training data, a convolutional layer using a filter of the MQTT training data, a pooling layer, a fully connected (FC) layer, and an output layer. There are a wide range of CNN layer combinations from which to select [47]. The CNN structure used in this study is illustrated in Figure 6.

$$x_i = f(w_i \otimes x_{i-1} + b_i), \quad (2)$$

where x_i is the IoT network data received from convolution filter l, i is the convolution kernel of the CNN layer, and $\otimes$ is the convolution operation. To transfer the output from the CNN layer to obtain the final output, the activation function $f(x)$ was used. The main goal of convolutional is to extract significant features from the data it receives. Convolution kernels can comprise multiple layers used to filter input data, each of which corresponds to a different weight and deviation coefficient [48]. The weighted parameter values indicate w_i, and the b_i is bias values. The following is an expression of the convolution process:

$$f(x) = tanh(x) = \frac{2}{1 + e^{-2x}} - 1, \quad (3)$$

where *tanh* is the function and x is the training input data. X_i is the output obtained from the CNN filters I, which denotes the convolution process, and the activation function is denoted by $f(x)$. A relatively large unit (ReLU) was chosen for the activation function of the convolutional layer. At the time of the model's implementation, we used all activation functions, such as sigmoid and tanh, among others. We found that the ReLU activation function was appropriate for processing our data, as it showed faster model training and a

more efficient prevention of gradient disappearance throughout the training process. In what follows, we describe the expression of the ReLU.

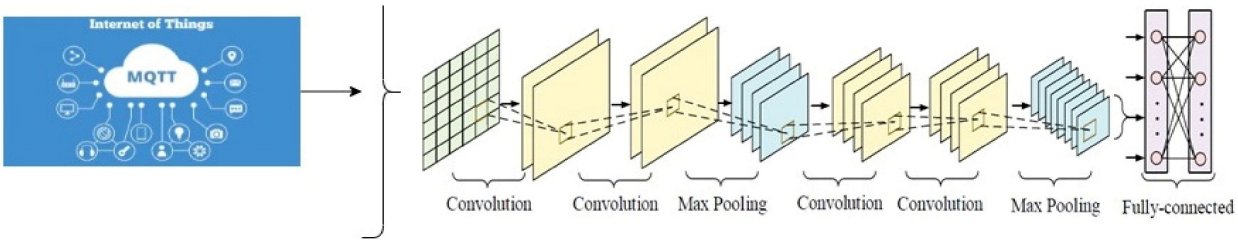

Figure 6. Structure of convolutional neural networks.

The primary objective of the max pooling layer is to achieve invariance and minimize the complexity of the convolution layer of the CNN by removing redundant information through down sampling and reducing the number of layers. To complete pooling, there are two basic methods: averaging pooling and maximum pooling. Average pooling, the most common method, involves selecting the average value in the computing area as the pooling result of the area, whereas max pooling involves selecting the maximum values in the computing area as the pooling result of the area. Since maximum pooling can retain more vital information than average pooling, the maximum pooling method was built on the CNN model. Maximum pooling can be defined as follows:

$$Q_j = Max\left(P_j^0, P_j^1, P_j^2 P_j^3 \ldots . P_j^t\right), \tag{4}$$

where Q_j is the output from the IoT cybersecurity dataset, j is the pooling area, Max is the operation, and P_j^t is the element of the pooling region j's element t.

FC layers serve as "classifiers" throughout the CNN. The FC layer is used to map the feature values from the filter and max pool layers into one layer. When overfitting occurs in the FC layer, the dropout method is used to randomly remove neurons to prevent the occurrence of overfitting.

The recurrent neural network (RNN) is the most well-known model for training temporal data. However, the standard RNN is difficult to train because of gradient explosion and disappearance. The LSTM model [43] is used to alleviate these problems by substituting units with a memory function with those buried in the RNN. Due to its modest weight, which changes over time, the LSTM model possesses long-term memory. Figure 7 depicts the structure of the LSTM model used in this study. The model receives its core information through the horizontal line. It is based on important gates, such as the forget gate, input gate, and output gate, which are used to store and retrieve previous knowledge and learn new information, and it performs this process using new information.

$$h_t = \text{sigma}\ \left(W_{xt} + U h_{t-1} + b^{(h)}\right), \tag{5}$$

$$f_t = \text{sigma}\ \left(W^{(f)} + X_t + U^{(f)}\ h_{t-1} + b^{(f)}\right), \tag{6}$$

where h_t is the input data for the current cell, h_{t-1} is the output data, f_t is the forget gate, $W^{(f)}$ is the weight, and $b^{(f)}$ is the bias. Sigma and tanh functions were employed to update the information in the input gate. Sigma was used to decide which data needed to be updated, whereas *tanh* was used to produce data that needed to be updated.

$$i_t = \text{sigma}\ \left(W^{(i)} + X_t + U^{(i)}\ h_{t-1} + b^{(i)}\right), \tag{7}$$

$$m_t = \tan \text{h}\ \left(W^{(m)} + X_t + U^{(m)}\ h_{t-1} + b^{(m)}\right), \tag{8}$$

$$c_t = i_t \cdot m_t + f_t \cdot c_{t-1}, \tag{9}$$

where c_{t-1} is the cell state; $f_t \cdot c_{t-1}$ and $i_t \cdot m_t$ must be combined to produce the next cell state when utilizing the preceding cell's state to update c_t, as the new information must be discarded.

$$o_t = \text{sigma}\left(W^{(o)} + X_t + U^{(o)}\, h_{t-1} + b^{(o)}\right), \tag{10}$$

$$h_t = o_t \cdot \tan \text{h}\,(c_t). \tag{11}$$

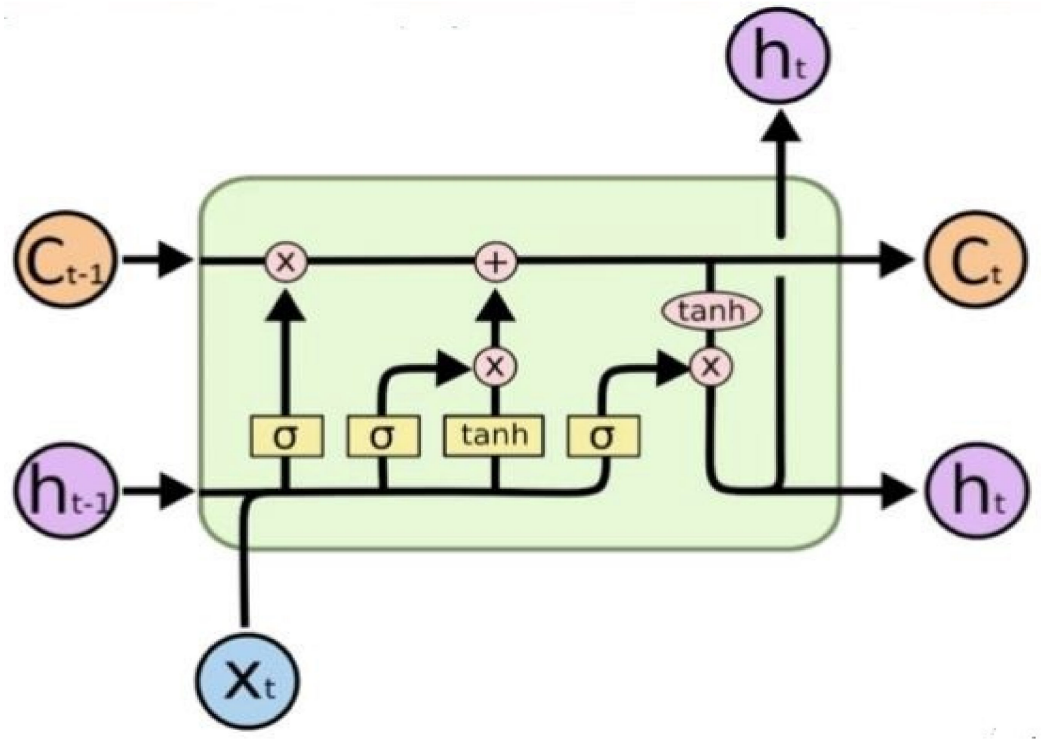

Figure 7. LSTM model's structure.

The information filtering during the learning step is called forget gate. It has two inputs, h_t and x_t, and the current cell state is denoted by c_t; 1 indicates that the cell was totally retained, while 0 indicates that it was completely dismissed. The following is a precise expression of the forget gate in terms of its function:

The architecture of the CNN-LSTM intrusion detection model proposed in this study is shown in Figure 8. The model was built primarily by processing the features of the CNN and LSTM models, such as fusion components. Numerical processing and normalization of the input were performed in the data preprocessing component to meet the criteria of the neural networks. There were three main layers in the CNN component: the convolutional layer with 64 filters and 128 filters, the max pooling layer, and the FC layer. Its primary job was to extract the important features from the dataset and determine whether the feature distribution of information about the intrusion detection system classified as normal or attacks. Where the component LSTM model are LSTM cells, and it is mostly used to detect intrusion information through the use of its memory function, which is a feature of this component. To produce the final result, the feature fusion of the CNN and LSTM model components were constructed of multilayer perceptrons (MLPs), whose primary goal was to use the link between the CNN and LSTM models to retrieve the data from the CNN and LSTM components, as well as to normalize what is considered for classification probability obtained from the CNN component. Table 2 shows the important parameters of CNN-LSTM.

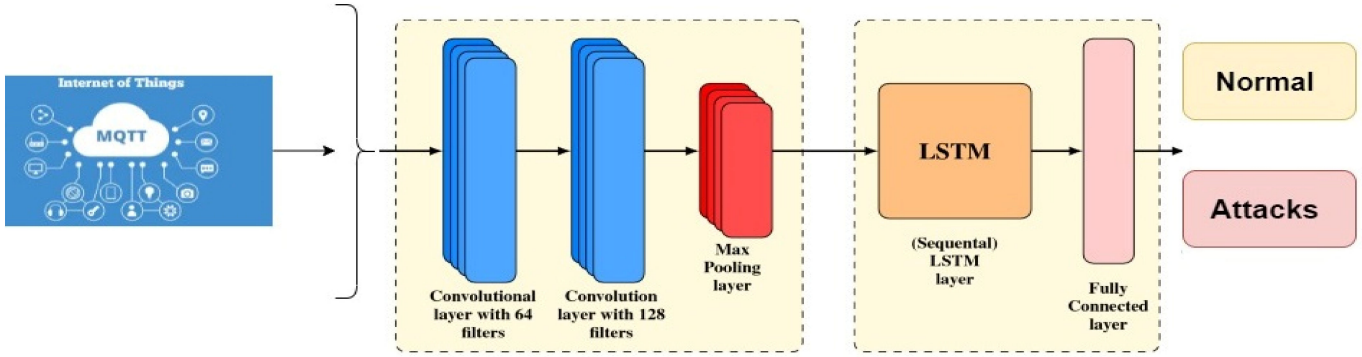

Figure 8. Architecture CNN-LSTM model.

Table 2. Parameter values of the CNN-LSTM approach.

Parameters Name	Significant Values
Convolution kernel size	3
Max pooling	5
Drop out of layer	0.50
FC layer	128
Activation function	ReLU
Optimizer function	Adam
Epochs	20
Batch size	120

2.4. Performance Measurements

The suggested algorithms' efficiency in identifying MQTT protocol IoT attacks was evaluated using the following statistical measures: accuracy, specificity, sensitivity, precision, and F score. The following are the equations for the parameters in question:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \quad (12)$$

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (13)$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (14)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (15)$$

$$F\ score = \frac{2 \times Precision \times Sensitivity}{Precision + Sensitivity} \times 100\% \quad (16)$$

3. Experiments

The proposed system was investigated using the MQTT protocol connected by using a number of IoT applications. The experiments were conducted using Python. The empirical results obtained from the proposed system were tested using various evaluation indicators.

3.1. MQTT Data Splitting

The MQTT dataset was divided such that 70% was used for training on the MQTT network traffic, and 30% was used to test the data and examine the results of the training process, which were then used to develop the model and test it as unseen data. Table 3 shows the volumes of the MQTT dataset.

Table 3. Volume of datasets.

Datasets	Volume of Data	Training Process	Testing Process
MQTT	330,936	264,748	66,188

3.2. Experimental Environments

To develop a security system, a robust environment is needed. The platforms used to run the proposed system are displayed in Table 4.

Table 4. The system's environmental preconditions and requirements.

Hardware Requirements	Software Requirements
RAM 16 GB CPU	Used Python 3.8 Used Numpy Version 1.19.3 Used TensorFlow Version 2.12 Used Keras Library 3.8

3.3. Results

ML and DL models were shown to be particularly efficient for detecting MQTT attacks in IoT environments. In this study, a standard MQTT dataset was employed to develop a cyberattack system. The MQTT dataset had 29 features, which contained 330,936 value injections of five attacks and normal.

3.3.1. Results of Machine Learning Models

The results of the KNN models are shown in Table 5. The KNN algorithm achieved an accuracy of 80.82%. We observed that the KNN showed good performance in detecting attacks. The percentage-weighted average metric of the KNN model was 86%, 81%, and 82% for both classes with respect to all performance indicators.

Table 5. Empirical results for the k-nearest neighbors algorithm.

Evaluation Metric	Precision (%)	Recall (%)	F1-Score (%)	Support	Time/s
Normal label	100	62	76	49,627	135.6
Attacks label	72	100	84	49,654	
Accuracy %	80.82				
Weighted average %	86	81	82	99,281	

The results of the LDA approach are shown in Table 6. Given the LDA algorithm's performance accuracy, the LDA model is not suitable for identifying MQTT protocol IoT attacks. The obtained results are modest, as the network dataset contained nonlinear data that presented challenges in finding the patterns.

Table 6. Empirical results for the linear discriminant analysis approach.

Evaluation Metric	Precision (%)	Recall (%)	F1-Score (%)	Support	Time/s
Normal label	100	53	70	41,327	180
Attacks label	68	100	81	41,407	
Accuracy %	76.72				
Weighted average %	84	77	75	99,281	

Figure 9 shows the confusion metrics of the KNN and LDA approaches for detecting MQTT attacks. The confusion metrics used four indicators—FP, FN, TP, and TN—to measure the proposed system's performance. The results of the KNN model show that 50.01% was true positive, that is, correctly classified data; the proportion of misclassification (false positive) was 0.00%, and the true negative, which refers to data classified as normal, was 30.81%. We observed that the false negative was very high, at 19.17%. Although the results of the LDA show a true positive proportion of 50.05%, the proportion of true negatives was much lower, at 26.67%, and the false negatives were very high, at 23.28%. Overall, the model had a 0.00% misclassification rate, which suggests that the ML model is effective for predicting assaults on the MQTT protocol.

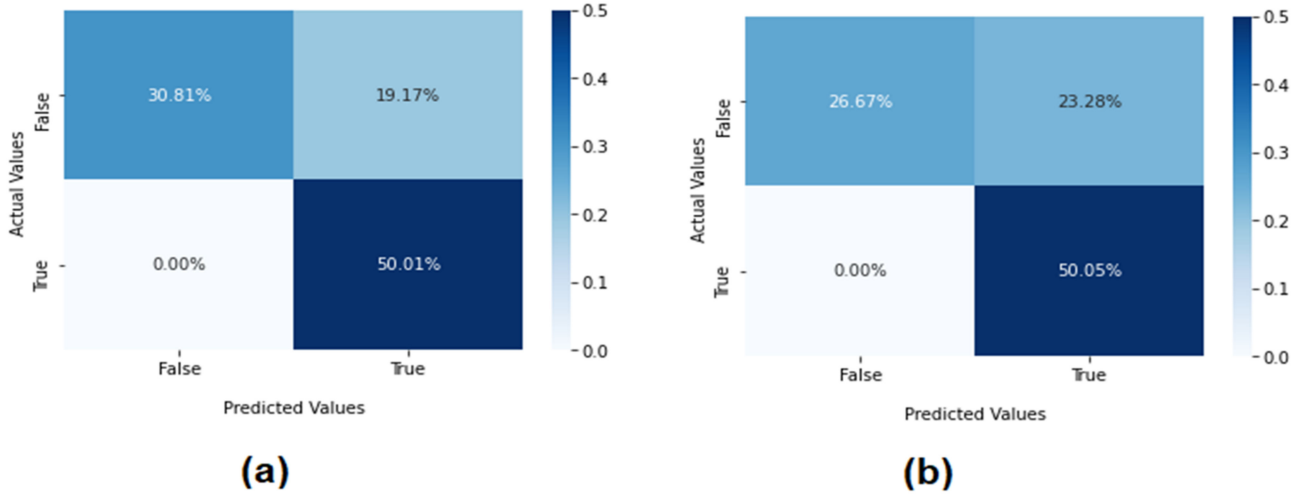

Figure 9. The confusion metrics for (**a**) the KNN method and (**b**) the LDA method.

3.3.2. Deep Learning Results

The empirical results of the CNN and CNN-LSTM are presented in Table 7. The MQTT dataset was divided into 70% as a training process for checking the ability of models to detect attacks, and 30% as testing for examining the DL models. An accuracy rate of 98.94% was reached by the CNN-LSTM model compared to 80.28% for the CNN model. Therefore, the CNN-LSTM model is appropriate for detecting MQTT protocol IoT intrusions. The main advantage of the CNN-LSTM model was that we combined two models to achieve good accuracy.

Table 7. Empirical results of the CNN and CNN-LSTM deep learning models.

Deep Learning Model	Loss Accuracy	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time/s
CNN	0.29	80.28	96.51	62.96	76.20	69.6
CNN-LSTM	0.11	98.94	99.27	99.07	99.17	52.02
		Cross validation 5 fold				
CNN		77.68	95.33	66.02	75.26	150.09
CNN-LSTM		93.22	95.68	96.12	95	130.30

The accuracy performance and loss validation of the CNN model is presented in Figure 10. We observed that the training and validation performances reached 81% with 200 epochs, with the training and testing loss decreasing to 0.299.

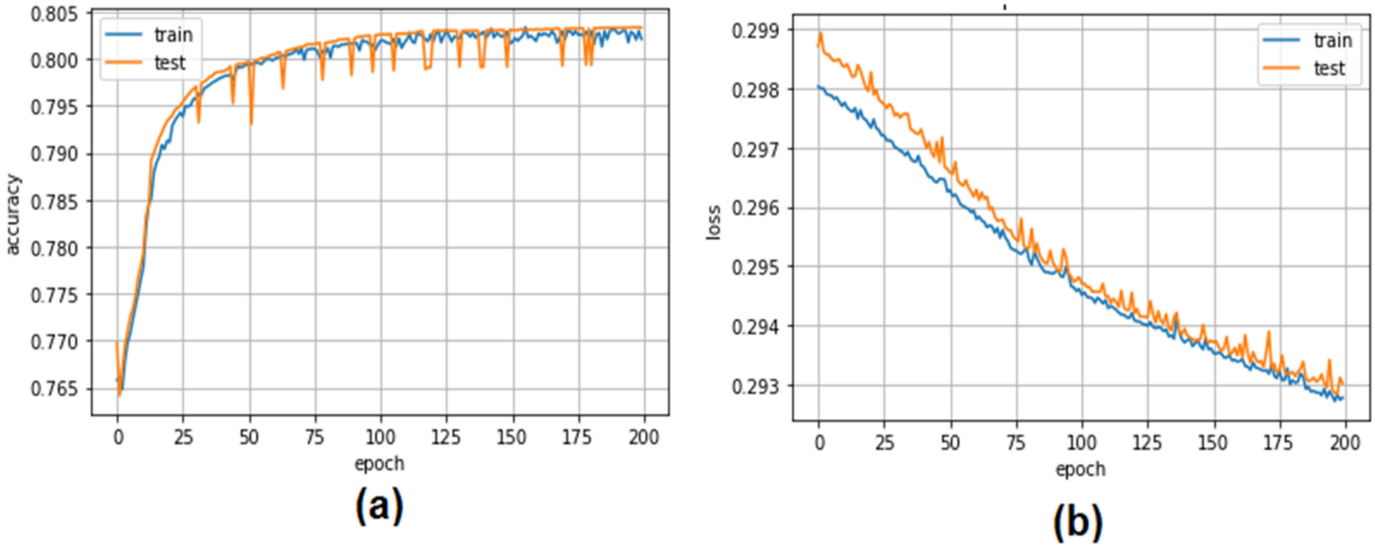

Figure 10. Performance of the CNN model in terms of (**a**) accuracy and (**b**) loss.

Figure 11 shows the accuracy performance and accuracy loss of CNN-LSTM during the training and validation phases. The performance of the CNN-LSTM model in the training phase reached 100%, with validation accuracy increasing to 98.94%. The accuracy loss of CNN-LSTM was very low, at 0.11. In general, the CNN model's performance in detecting MQTT protocol IoT intrusions was strong.

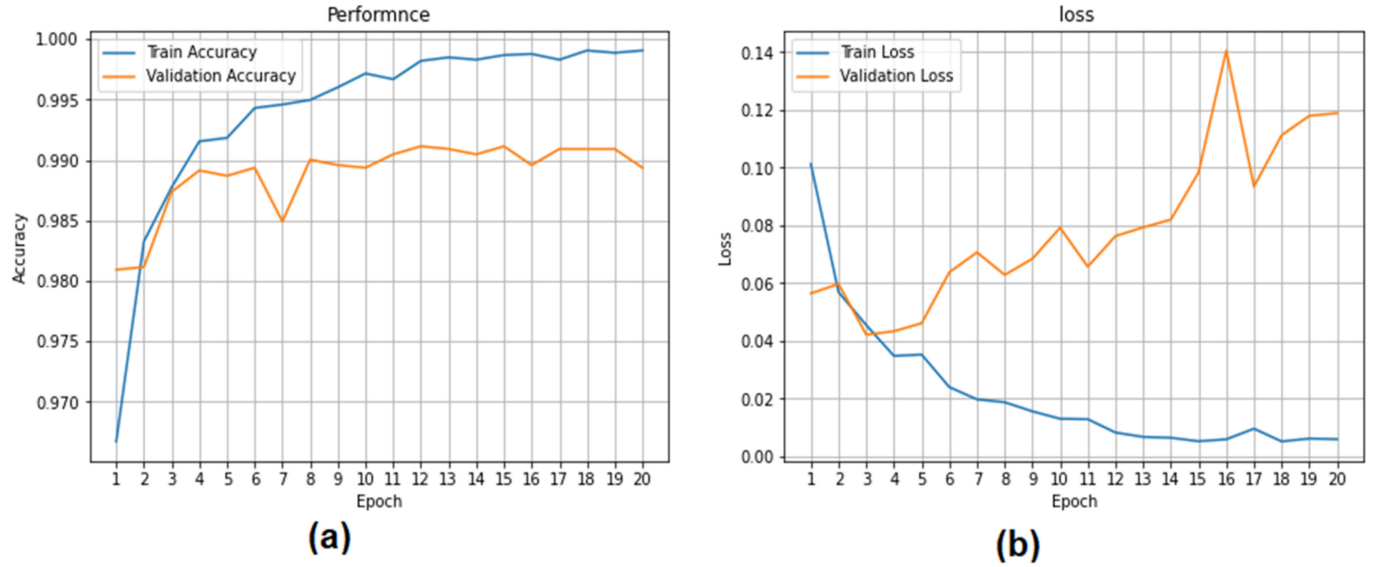

Figure 11. Performance of the CNN-LSTM model in terms of (**a**) accuracy and (**b**) loss.

In addition, the mean absolute error statistical analysis (MAE), the mean squared error statistical analysis (MSE), the root mean squared error statistical analysis (RMSE), and the R^2 were utilized in order to calculate the percentage error that existed between the target values and the predictions. The statistical analysis of ML and DL using binary classification is broken down and summarized in Table 8. According to the results of the CNN-LSTM model, the correlation between the target and the prediction values was the maximum it could be (R^2 = 98.42%), and the prediction error MSE = 0.0032.

Table 8. Statically analysis.

Models	MAE	MSE	RMSE	R^2 (%)
LDA	0.232	0.232	0.4824	76.72
KNN	0.1917	0.1917	0.4824	80.82
CNN	0.043	0.092	0.098	95.16
CNN-LSTM	0.0032	0.0032	0.0068	98.42

Table 9 provides a comprehensive analysis of the recommended ML and DL models by displaying the outcomes of the proposed approaches in comparison to other previous systems. These findings may be found in the Table 7. We obtained the conclusion that the accuracy of the CNN-LSTM model was 98.94%, which is a very high percentage. Both the RF tree and the LSTM demonstrated pretty respectable degrees of accuracy. Figure 12 offers a graphical depiction of the primary results achieved by the proposed algorithms in contrast to a number of the most common practices already in use.

Table 9. Results of the proposed system against existing security systems using the same datasets.

Reference	Year	Datasets	Model	Accuracy
Ref. [48]	20121	Same dataset	LSTM	97.13%
Ref. [49]	2021	Same dataset	Deep learning	97.09%
Proposed model	2022	Same dataset	CNN-LSTM	98.94%

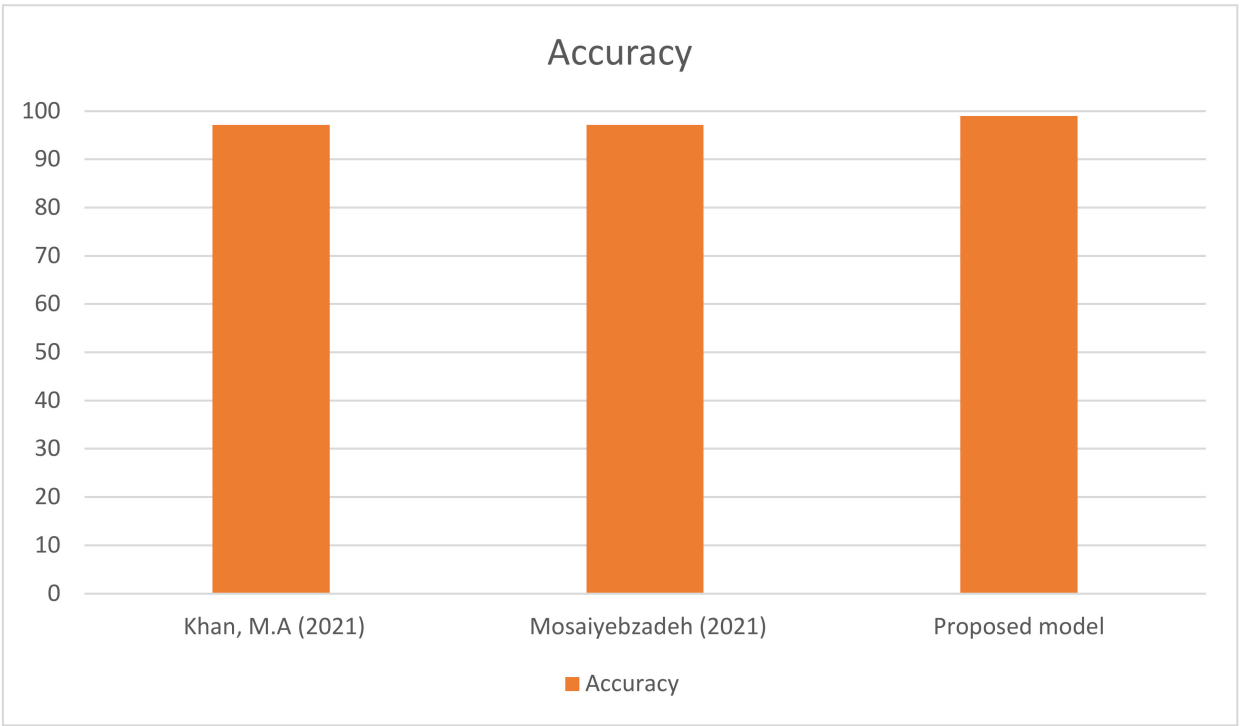

Figure 12. Comparison of proposed CNN-LSTM against security systems [48,49].

4. Conclusions

The considerable growth in the volume of communications resulted in an increase in vulnerability, resulting in a variety of intrusions that place the integrity of the proposed security system at risk. Depending on the nature of each attack, a variety of repercussions may occur, including the introduction of malware that may cause damage to equipment,

unauthorized access to network information, and DoS attacks. As a result, advanced artificial intelligence algorithms capable of detecting these attacks are being developed to maintain the integrity of IoT platforms. Detecting DoS attacks in MQTT networks using ML and DL methods was the focus of this study. These techniques were used to defend the MQTT network in an IoT context, based on the KNN and LDA methods, as well as the CNN-LSTM model. Standard MQTT network information obtained from various IoT contexts was used to create the security solution. The KNN and LDA methods, two ML algorithms, proved to be effective at detecting MQTT network attacks.

The hybrid CNN-LSTM model efficiently anticipated MQTT network breaches. Because it achieved a high degree of precision (98.94%), the CNN-LSTM model is particularly successful at identifying intrusions in IoT environments. The CNN-LSTM model showed great accuracy in identifying malicious attacks on IoT devices based on the output results of the ML and DL methods. Overall, the suggested technique was effective at identifying malicious attacks on the MQTT network. The scope of the work carried out so far is limited to the MQTT protocol attack detection. Therefore, in the future, authors can apply more advanced DL algorithms and feature selection approaches to further improve the security system. The developing system does not handle all of the MQTT vulnerabilities that were found up to this point, which is another limitation of this endeavor. In addition, one of our goals is to examine the susceptibility of a variety of Internet of Things protocols to fresh kinds of assaults. Our goal is to come up with an original model for emerging vulnerabilities that is based on deep learning.

Author Contributions: T.H.H.A. and A.A. resources; T.H.H.A. data curation, A.A.; writing—original draft preparation, T.H.H.A. and A.A.; writing—review and editing, A.A.; visualization, T.H.H.A. and A.A.; supervision, T.H.H.A.; project administration, T.H.H.A. and A.A.; funding acquisition, T.H.H.A. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were funded by the Deanship of Scientific Research at King Faisal University for the financial support under grant No. NA000245.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This work was supported through the Annual Funding track by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [NA000245].

Conflicts of Interest: The authors declare no conflict of interest.

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. [CrossRef]
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. Atmoko, R.A.; Yang, D. Online Monitoring & Controlling Industrial Arm Robot Using mqtt Protocol. In Proceedings of the 2018 IEEE International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (Robionetics), Bandung, Indonesia, 8–10 August 2018; pp. 12–16.

8. Safaei, B.; Monazzah, A.M.H.; Bafroei, M.B.; Ejlali, A. Reliability Side-Effects in Internet of Things Application Layer Protocols. In Proceedings of the 2017 2nd International Conference on System Reliability and Safety (ICSRS), Milan, Italy, 20–22 December 2017; pp. 207–212.
9. Alkahtani, H.; Aldhyani, T.H.H. Artificial Intelligence Algorithms for Malware Detection in Android-Operated Mobile Devices. *Sensors* **2022**, *22*, 2268. [CrossRef] [PubMed]
10. Thantharate, A.; Beard, C.; Kankariya, P. CoAP and MQTT Based Models to Deliver Software and Security Updates to IoT Devices over the Air. In Proceedings of the 2019 International Conference on Internet of Things (iThings), Los Alamitos, CA, USA, 14–17 July 2019; pp. 1065–1070. [CrossRef]
11. Rayes, A.; Salam, S. *Internet of Things from Hype to Reality—The Road to Digitization*, 2nd ed.; Springer: Cham, Switzerland, 2019.
12. Belli, L.; Cilfone, A.; Davoli, L.; Ferrari, G.; Adorni, P.; Nocera, F.D.; Dall'Olio, A.; Pellegrini, C.; Mordacci, M.; Bertolotti, E. IoT-Enabled Smart Sustainable Cities: Challenges and Approaches. *Smart Cities* **2020**, *3*, 52. [CrossRef]
13. Rehma, A.A.; Awan, M.J.; Butt, I. Comparison and Evaluation of Information Retrieval Models. *VFAST Trans. Softw. Eng.* **2018**, *6*, 7–14.
14. Alam, T.M.; Awan, M.J. Domain analysis of information extraction techniques. *Int. J. Multidiscip. Sci. Eng.* **2018**, *9*, 1–9.
15. Koo, J.; Kang, G.; Kim, Y.-G. Security and Privacy in Big Data Life Cycle: A Survey and Open Challenges. *Sustainability* **2020**, *12*, 10571. [CrossRef]
16. Privalov, A.; Lukicheva, V.; Kotenko, I.; Saenko, I. Method of Early Detection of Cyber-Attacks on Telecommunication Networks Based on Traffic Analysis by Extreme Filtering. *Energies* **2019**, *12*, 4768. [CrossRef]
17. Nishanth, N.; Mujeeb, A. Modeling and detection of flooding-based denial-of-service attack in wireless ad hoc network using Bayesian inference. *IEEE Syst. J.* **2020**, *15*, 17–26. [CrossRef]
18. Gupta, M.; Jain, R.; Arora, S.; Gupta, A.; Awan, M.J.; Chaudhary, G.; Nobanee, H. AI-enabled COVID-19 Outbreak Analysis and Prediction: Indian States vs. Union Territories. *Comput. Mater.* **2021**, *67*, 933–950. [CrossRef]
19. 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. [CrossRef]
20. Ali, Y.; Farooq, A.; Alam, T.M.; Farooq, M.S.; Awan, M.J.; Baig, T.I. Detection of Schistosomiasis Factors Using Association Rule Mining. *IEEE Access* **2019**, *7*, 186108–186114. [CrossRef]
21. Javed, R.; Saba, T.; Humdullah, S.; Jamail, N.S.M.; Awan, M.J. An Efficient Pattern Recognition Based Method for Drug—Drug Interaction Diagnosis. In Proceedings of the 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; pp. 221–226.
22. Nagi, A.T.; Awan, M.J.; Javed, R.; Ayesha, N. A Comparison of Two-Stage Classifier Algorithm with Ensemble Techniques on Detection of Diabetic Retinopathy. In Proceedings of the 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; pp. 212–215.
23. Abdullah, A.; Awan, M.; Shehzad, M.; Ashraf, M. Fake News Classification Bimodal Using Convolutional Neural Network and Long Short-Term Memory. *Int. J. Emerg. Technol. Learn.* **2020**, *11*, 209–212.
24. 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. [CrossRef]
25. 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. [CrossRef]
26. 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. [CrossRef]
27. 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.
28. 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.
29. 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.
30. 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.
31. 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. [CrossRef]
32. 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. [CrossRef]
33. 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.

34. 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. [CrossRef]
35. 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. [CrossRef]
36. 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.
37. 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. [CrossRef]
38. 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. [CrossRef]
39. Ahuja, N.; Singal, G.; Mukhopadhyay, D.; Kumar, N. Automated DDOS attack detection in software defined networking. *J. Netw. Comput. Appl.* **2021**, *187*, 103108. [CrossRef]
40. 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. [CrossRef]
41. Dehkordi, A.B.; Soltanaghaei, M.; Boroujeni, F.Z. The DDoS attacks detection through machine learning and statistical methods in SDN. *J. Supercomput.* **2021**, *77*, 2383–2415. [CrossRef]
42. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1153–1176. [CrossRef]
43. Mishra, P.; Varadharajan, V.; Tupakula, U.; Pilli, E.S. A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 686–728. [CrossRef]
44. Soucy, P.; Mineau, G.W. A Simple KNN Algorithm for Text Categorization. In Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 29 November–2 December 2001; pp. 647–648.
45. Deng, Z.; Zhu, X.; Cheng, D.; Zong, M.; Zhang, S. Efficient kNN classification algorithm for big data. *Neurocomputing* **2016**, *195*, 143–148. [CrossRef]
46. Zheng, D.; Hong, Z.; Wang, N.; Chen, P. An Improved LDA-Based ELM Classification for Intrusion Detection Algorithm in IoT Application. *Sensors* **2020**, *20*, 1706. [CrossRef]
47. Vaccari, I.; Chiola, G.; Aiello, M.; Mongelli, M.; Cambiaso, E. MQTTset, a New Dataset for Machine Learning Techniques on MQTT. *Sensors* **2020**, *20*, 6578. [CrossRef]
48. Khan, M.A.; Khan, M.A.; Jan, S.U.; Ahmad, J.; Jamal, S.S.; Shah, A.A.; Pitropakis, N.; Buchanan, W.J. A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT. *Sensors* **2021**, *21*, 7016. [CrossRef]
49. Mosaiyebzadeh, F.; Rodriguez, L.G.A.; Batista, D.M.; Hirata, R. A Network Intrusion Detection System using Deep Learning against MQTT Attacks in IoT. In Proceedings of the 2021 IEEE Latin-American Conference on Communications, Santo Domingo, Dominican Republic, 17–19 November 2021; pp. 1–6. [CrossRef]