IoT Protocol-Enabled IDS based on Machine Learning

Rehab Alsulami
Cybersecurity Department, CCSE, University of Jeddah, Saudi Arabia
rehabayed.cs@gmail.com

Batoul Alqarni
Cybersecurity Department, CCSE, University of Jeddah, Saudi Arabia
batoull88xx@gmail.com

Rawan Alshomrani
Cybersecurity Department, CCSE, University of Jeddah, Saudi Arabia
rwx.rwn@gmail.com

Fatimah Mashat
Cybersecurity Department, CCSE, University of Jeddah, Saudi Arabia
famashat2000@gmail.com

Tahani Gazdar
Cybersecurity Department, CCSE, University of Jeddah, Saudi Arabia
taalgazdar@uj.edu.sa (corresponding author)

Received: 21 September 2023 | Revised: 23 October 2023 | Accepted: 4 November 2023

 DOI: https://doi.org/10.48084/etasr.6421

ABSTRACT

During the last decade, Internet of Things (IoT) devices have become widely used in smart homes, smart cities, factories, and many other areas to facilitate daily activities. As IoT devices are vulnerable to many attacks, especially if they are not frequently updated, Intrusion Detection Systems (IDSs) must be used to defend them. Many existing IDSs focus on specific types of IoT application layer protocols, such as MQTT, CoAP, and HTTP. Additionally, many existing IDSs based on machine learning are inefficient in detecting attacks in IoT applications because they use non-IoT-dedicated datasets. Therefore, there is no comprehensive IDS that can detect intrusions that specifically target IoT devices and their various application layer protocols. This paper proposes a new comprehensive IDS for IoT applications called IP-IDS, which can equivalently detect MQTT, HTTP, and CoAP-directed intrusions with high accuracy. Three different datasets were used to train the model: Bot-IoT, MQTT-IoT-IDS2020, and CoAP-DDoS. The obtained results showed that the proposed model outperformed the existing models trained on the same datasets. Additionally, the proposed DT and LSTM models reached an accuracy of 99.9%.

Keywords-IDS; IoT; DT;LSTM

I. INTRODUCTION

The Internet of Things (IoT) refers to a network of physical items integrated with sensors, software, and other technologies that connect to and exchange data with other devices and systems through the Internet. Such devices range from common domestic items to complex industrial machines. IoT devices may be secure at the time of purchase but become vulnerable when hackers discover new security flaws or bugs. An Intrusion Detection System (IDS) is hardware or software that scans a network for harmful activity or potential security attacks and alerts the administrator. The IDS collects data that contain evidence of an attack using gathering models. After processing the data, an analysis module discovers the attacks and reports them to the administrator [1]. There are two IDS approaches: anomaly-based and signature-based. Signature-based IDS have a repository of attack signatures to compare with network data, and if a match is discovered, an alarm is triggered. Although this approach is often very effective in recognizing known threats, it is less effective against mutations from existing and zero-day attacks [2]. Anomaly-based IDSs use Machine Learning (ML) models to recognize attacks. ML

models define the baseline of system activities and/or attack patterns, and any system activity that deviates significantly is reported [2]. Traditional IDSs designed for conventional networks are less effective in the context of IoT systems due to the specific vulnerabilities and risks that threaten this kind of application. The resource-constrained nature of IoT nodes, in addition to the large-scale and time-constrained applications in the IoT context, requires new customized protocols to provide the required QoS. New protocols are used, such as Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP), in addition to the de-facto HTTP. All these protocols have many vulnerabilities that require specific care [2].

The MQTT protocol is a machine-to-machine and publish-subscribe protocol that provides reliable and secure communications among IoT nodes and is considered the most popular protocol used in messaging. Many vulnerabilities in this protocol have emerged and are reported to be increasing, especially in the recent period from 2014 to 2020. One of the existing vulnerabilities is the insufficient check of packet length before parsing, which makes attackers exploit it to perform buffer overflow, Denial of Service (DoS), remote code execution, and reading memory contents. Another vulnerability is the ignorance of the required validation field that could cause a server crash, buffer overflow, remote code execution, or broker crash. The MQTT protocol also suffers from insufficient logical error checks that could lead to DoS, code execution, server crashes, or buffer overflow attacks [3].

CoAP is a web-based protocol dedicated to constrained nodes and networks such as IoT. Unfortunately, it has many serious weaknesses, such as the improper incoming parsing of messages that could lead to remote code execution, which can affect the availability of the CoAP node. In addition, in the CoAP context, improper implementation of proxies and cache access control might lead to compromise of their content and threaten the integrity and confidentiality of CoAP messages. Additionally, the improper configuration of new CoAP nodes might result in unauthorized access to the CoAP environment by unauthorized nodes. One more vulnerability for CoAP is the unreliable generation of cryptographic keys that could compromise the nodes. Another important weakness is the IP address spoofing of CoAP nodes, which allows an attacker to generate spoofed response messages and acknowledgments as well as reflection/amplification attacks [4].

Similarly, the HTTP protocol has some major vulnerabilities. SQL injection is one of the most prevalent types of web application security vulnerabilities. In addition, many security issues associated with managing a user's identity can be caused by faulty authentication and session management in HTTP. Security misconfiguration is another vulnerability in web applications. ML approaches have recently been presented as successful techniques for detecting network attacks, especially IoT networks [1, 5-6]. In the context of intrusion detection, ML models are known for their ability to detect zero-day attacks. Distinct efforts have been made to design ML-based IDSs for IoT. They provide many brilliant solutions, but unfortunately, there is no consideration of specific IoT protocols, such as MQTT and CoAP. Few existing IDSs can detect forged or invalid packets in IoT communication based on specific protocols, such as MQTT or CoAP. Furthermore, in existing IDSs, not all ML models are trained using IoT datasets, and therefore their performance may be reduced once deployed in an IoT environment [7].

Motivated by the widespread application of ML and its proven efficiency in detecting zero-day vulnerabilities, many studies adopted ML to design anomaly-based IDSs [1]. In [8], a Deep Neural Network (DNN)-based model was proposed to detect attacks that exploit vulnerabilities in the MQTT protocol. The performance of the proposed model was compared with conventional ML algorithms, using a dataset that contained records on three different types of attacks, including man-in-the-middle, intrusion, and DoS. The results showed that the DNN model performed better than Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The proposed model was good enough to detect MQTT-directed intrusions, yet it was limited to detecting only such intrusions. In [9], an IDS was proposed to detect intrusions using the MQTT protocol, comparing many shallow and Deep Learning (DL) models, and the results showed that the Decision Tree (DT) outperformed the DL models. Similarly, in [10] a model was proposed to detect man-in-the-middle attacks that exploit MQTT vulnerabilities. Many classification techniques were used, such as Non-Convex Boundary over Projections (NCBoP), Approximate Convex Hull (ACH), K-Means, and Principal Component Analysis (PCA), on a real dataset. The obtained results showed that PCA was the best approach for detecting man-in-the-middle attacks. Unfortunately, this study focused only on evaluating the performance of different techniques in terms of accuracy and shortest training time for MQTT datasets and did not consider other attack types.

In the literature, only one study focused on proposing an ML-based model for intrusion detection in CoAP [11]. This study proposed an IDS for the detection and prevention of attacks against Internet-integrated CoAP communication environments, putting into practice and evaluating the efficiency of anomaly-based intrusion detection using an SVM model to detect DoS attacks against the CoAP protocol. The main weakness of this model was that it focused on only one protocol and one attack. In [12], many ML and DL models were proposed, including SVM, Random Forest (RF), DT, and Recurrent Neural Networks (RNNs) that concentrated on identifying DoS and Distributed DoS (DDoS) attacks in the transport and application layers using the BoT-IoT dataset. The model achieved an average accuracy of more than 99%, yet this model was not comprehensive enough to address threats on IoT networks since it only considered DoS/DDoS attacks. In [13], a DL model was proposed to recognize intrusions in IoT networks using the BoT-IoT dataset and ML algorithms, such as RF, and DL techniques such as Convolutional Neural Networks (CNNs) for the classification of attacks. The results showed that the DL models had higher DoS, DDoS, reconnaissance, and data theft detection capabilities than the ML models. However, the proposed models did not consider detecting attacks in application protocols but only in the network layer. Similarly, in [14], network layer attacks in the IoT environment were detected using the CICID2017 dataset, which is a generic dataset not dedicated to IoT networks.

Although the proposed model successfully detected many attacks, it may not be effective in the IoT context.

In [15], many shallow and DL models were used to design an IDS for IoT. The results showed that the shallow models outperformed the DL models in terms of accuracy for all devices. In [16], an IDS with a Deep Belief Network (DBN) was used to detect network-level attacks. The TON-IoT-Weather dataset [17] contains seven different attack types, such as DDoS, password cracking, scanning, etc, but not application layer attacks. In [18], a framework for intrusion detection in IoT was proposed, using three ML models trained on NSL-KDD datasets, but the models had very low accuracy and the dataset was not IoT-specific.

This study aims to build a new comprehensive anomaly-based network IDS, based on ML models, to help users monitor their IoT system by detecting intrusions specifically in IoT application layer protocols. The proposed system is called IP-IDS, which stands for IoT-Protocols enabled IDS. The proposed system aimed to overcome the drawbacks of traditional non-IoT-based ML approaches that suffer from low accuracy, inability to detect novel attacks, need for updates for every new attack pattern discovered, and lack of self-learning. IP-IDS aimed to improve ML-based solutions that assume that the application layer protocol is always HTTP, ignoring the vulnerabilities in the special lightweight protocols of IoT devices, such as MQTT and CoAP. Many studies proposed IDS solutions based on ML models for IoT, but as these models were not trained on IoT datasets, they produced inaccurate results [18]. Furthermore, existing ML-based IDS sought to detect standalone attacks, such as SQL injection, DoS, ransomware, XSS, etc [1, 12-13, 16]. Unlike these models, the scope of IP-IDS involved designing a comprehensive IDS that detects many types of intrusions in different application layer protocols, based on ML and trained on IoT-specific datasets. This study used IoT-dedicated datasets, such as MQTT-IoT-IDS2020 [19], CoAP-DDoS [20], and Bot-IoT [21]. The key contributions of this study are:

- Different ML/DL models for each of the specified IoT protocols are created: MQTT, CoAP, and HTTP.
- The models are trained and tested using specific IoT datasets.
- The proposed models are compared with the existing solutions.
- The best models in terms of performance are deployed in one comprehensive IDS.

II. THE PROPOSED IDS

A. *Architecture of IP-IDS*

This study proposes a new comprehensive IDS based on ML to accurately detect attacks on MQTT, CoAP, and HTTP using dedicated IoT datasets. As shown in Figure 1, the proposed IP-IDS consists of 3 main components: traffic collection, traffic analysis, and reporting. The traffic collection module is responsible for sniffing traffic flowing in the network and saving it in .pcap files. Traffic is then classified according to the application layer protocol into three classes, MQTT, CoAP, and HTTP, stored in separate files. Afterward, collected and classified traffic is fed into the analysis model to detect potential intrusions. For each application layer protocol, a different ML model was used to detect vulnerabilities related to this protocol. Each model classifies the traffic to determine whether it is normal (benign) or abnormal (attack) traffic and specifies the attack type. Once an attack is detected, a notification is sent to the administrator through a dashboard.

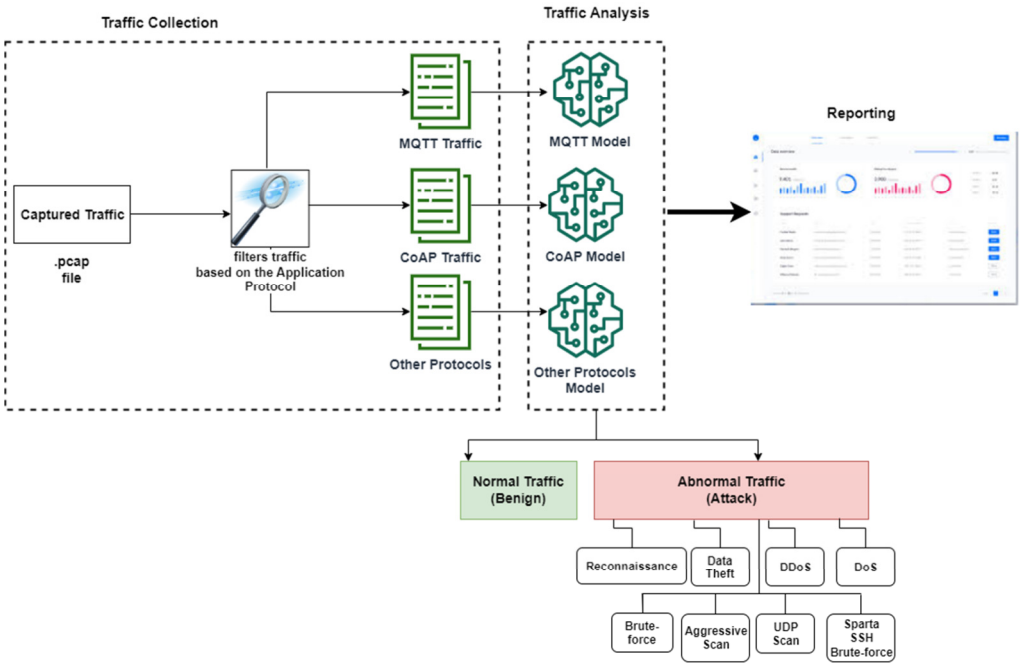

Fig. 1. Components of the IP-IDS.

B. Methodology

The development of IP-IDS was divided into four phases: Data collection, data prepossessing, modeling, and testing.

1) Datasets

Three datasets were used to train the ML/DL: MQTT-IoT-IDS2020 [19], CoAP-DDoS [20], and Bot-IoT [21]. Each one contains records specific to an application protocol and attacks that rely on it. The MQTT-IoT-IDS2020 dataset was created using a simulated sensor network that mimicked a real MQTT IoT network in a typical operating environment and contains both general networking scanning and MQTT brute-force attacks. The dataset is available in two formats: processed features and its original raw capture format (.pcap files). Characteristics are divided into three categories, i.e. bidirectional, packet-based, and unidirectional, and each set of features is used exclusively. Five scenarios make up the dataset. The first is for regular operation, and the other four are for attacks: aggressive scan, User Datagram Protocol (UDP) scan, Sparta SSH brute-force attack, and MQTT brute-force attack [19]. The second dataset was the CoAP-DDoS, which is used to classify DoS attacks on CoAP against IoT devices. It has 17 features per DoS attack, i.e. IP source, timestamp, ethernet type, and IP version, The dataset contains a total of 661,304 records that correspond to benign and malicious traffic [20]. The third dataset was the BoT-IoT, which is an IoT-dedicated dataset that includes both typical IoT-related and other network traffic, as well as numerous attack traffic types frequently launched by botnets. This dataset is labeled and collected from a realistic testbed. The label features an attack flow, an attack category, and an attack subcategory for multiclass classification use. The BoT-IoT dataset consists of 74 CSV files with more than 72,000,000 records, each with 46 features. The dataset contains 10 different types of attacks, including OS fingerprinting, DoS (HTTP, TCP, UDP), DDoS (HTTP, TCP, UDP), service scan, keylogging, and data theft [21].

2) Data Prepossessing

The Google Colaboratory Pro platform was used for implementation and experiments. The following libraries were used in the preprocessing of the datasets:

- Numby is a Python library that is used to perform a wide range of mathematical operations on arrays. It enhances Python with strong data structures that ensure effective calculations with arrays and matrices.
- Pandas is a Python library that is used in tasks related to ML and data science. It is built on top of Numpy, which supports multidimensional arrays.
- Os is a Python library that provides many functions to interact with the operating system.

The MQTT-IoT-IDS2020 dataset consists of five recorded scenarios: one normal operation and four attack scenarios (brute force, scan_a, scan_su, Sparta). Five CSV files were created, one for each class. In the four attack classes, any records that have a value of zero in the *is_attack* feature (label) were dropped. Then the five classes were concatenated into one CSV file and a column named class was added to label the records [9]. The dataset is not balanced, as most records correspond to the normal class, which has a total of 86,008 records, while the minority class has 14,116 records. However, the data imbalance negatively impacts the performance of the ML models. So, a resampling technique was applied, which resulted in a balanced dataset where each class had 30000 records. The sklearn.utils library was used to generate a random resampling from the dataset.

Feature selection can be used to minimize the generalization error and the complexity of ML models. This is a process that selects the most important features to improve computational efficiency. It is applied when there are redundant or irrelevant features, and is also known as feature dimensionality reduction [22]. This study used the RF feature selection model [22], which is characterized by better generalization and interpretation and high accuracy. Each feature is assigned an importance score according to how significant it is in predicting the class. Then, only the features that have an importance score above zero are kept. In this dataset, three features were dropped (*fwd_num_rst_flags*, *fwd_num_urg_flags*, and *bwd_num_urg_flags*), because their importance score was zero. Therefore, only 29 of 32 features were considered in the training and testing processes.

The CoAP-DDOS dataset consists of 624,938 records, of which 39.47% correspond to malicious records. The dataset is divided into time-based subsections. A subsection is considered malicious if it includes more than 350 packets exchanged between the two malicious IP addresses (192.168.1.12) and (92.168.1.5). Forty-two features were extracted from the packets and were then reduced to 16. The arrays of packets that carried the flows were asymmetrical to each other and padded to ensure that all labeled data had a consistent shape. The array of all packet values in each packet flow was also normalized using the Frobenius norm. Training and testing data were shuffled after splitting.

As the Bot-IoT dataset is imbalanced, it will be biased towards the majority class in the obtained results. The DDoS, which is the majority class, has 1,926,624 records. The theft, which is the minority class, has only 79 records. Similarly, there is a huge gap between the total number of records corresponding to malicious traffic (>=3.5 million records), while the benign class has only 477 records. Using the *resample* function from the sklearn.utils library, the dataset categories were resampled to 30,000 records for each to overcome the bias problem. Other data preprocessing techniques that were applied to this dataset were One-Hot Encoding, IP address cleaning, and Normalization. One-Hot Encoding was used to convert category, subcategory, and proto features into numerical dummy features. IP address cleaning was required to convert the string IP address from the form 192.168.100.7 to an integer number such as 1.144249e-35 using the *clean_ip* function from the "dataprep.clean" library. In addition, normalization was used to transform all values on a similar scale, which could improve the model's performance and training stability. Similarly to the MQTT-IoT-IDS2020 dataset, a feature selection was applied using the RF classifier to reduce the dimensionality of the BoT-IoT dataset.

Consequently, the *saddr_clean*, *proto_arp*, *daddr_clean*, and *proto_ipv6-ICMP* features were dropped due to their importance score being zero. Therefore, only 26 of 30 features were used in the training and testing processes.

3) Modeling

ML is a branch of Artificial Intelligence (AI) that focuses on giving systems the capacity to automatically learn and improve from experience and collected data [23]. DL is a type of ML that enables computers to learn by analyzing patterns through multiple layers of processing. DL algorithms are taught using huge volumes of labeled data and neural network topologies that learn features directly from data, enabling computers to perform better object detection [24]. To design IP-IDS, a model was needed to scan the traffic (input) and decide whether it is benign or malicious. In case of malicious traffic, the model should be able to accurately classify the attacks. DT and LSTM were trained using the above datasets to select the appropriate model to be deployed. DT and LSTM are two of the most widely used algorithms in intrusion detection [1]. DT is one of the most used supervised ML techniques that is applied to a provided dataset for both classification and regression, by organizing a series of rules in a tree structure. The model is organized like a typical tree, with nodes, branches, and leaves, and a feature is represented by each node. Each leaf indicates a potential result or class label, whereas a decision or a rule is represented by a branch. The DT algorithm automatically chooses the best features before performing pruning to remove unnecessary branches from the tree to prevent overfitting [7]. LSTM is a DL algorithm designed to overcome the long-term dependence issue. The primary characteristic of LSTM is its ability to store data or cell state for future use in the network. This attribute is useful for performing analysis on temporal data that change over time [1, 25].

III. EXPERIMENTS AND RESULTS

A. Experimental Setup

The MQTT-IoT-IDS2020 dataset was divided into 80% for training and 20% for testing. The BoT-IoT and CoAP-DDoS datasets were divided into 75% for training and 25% for testing. The LSTM model was implemented using the Tensorflow and Keras packages. The ADAM optimizer was used to optimize the categorical cross-entropy loss function in the case of multiclassification, and binary cross-entropy in the case of binary classification. For CoAP-DDoS, *sigmoid* and *tanh* were used as activation functions since this is a binary classification problem, and *sparse_categorical_crossentropy* as a loss function. The model was trained for 60 epochs. For MQTT and Bot-IoT, *softmax* was used as the activation function because it is a multiclassification problem. The LSTM output was equal to 100 and the number of epochs was set to 150. The loss function was the categorical cross-entropy for both datasets since the predicted labels are one-hot encoded.

B. Results

Figure 2 shows the average accuracy of DT and LSTM for the three datasets. Accuracy describes to which extent the model is exact in detecting a class among all classes, calculated as follows:

$$Accuracy = \frac{Correctly\ classified\ input}{Total\ number\ of\ inputs}$$

Figure 2 shows that DT performed better in terms of accuracy for all datasets compared to LSTM. Additionally, LSTM and DT had close accuracy for Bot-IoT and MQTT-IoT-IDS (approximately 99.99%). This high accuracy was due to the balance of the datasets and the feature selection technique used. Accuracy is affected by an uncertainty factor that relies on the unbalance of the dataset. The confusion matrix of LSTM in the MQTT-IoT-IDS dataset portrayed in Figure 3 shows that there is a small rate of false positives.

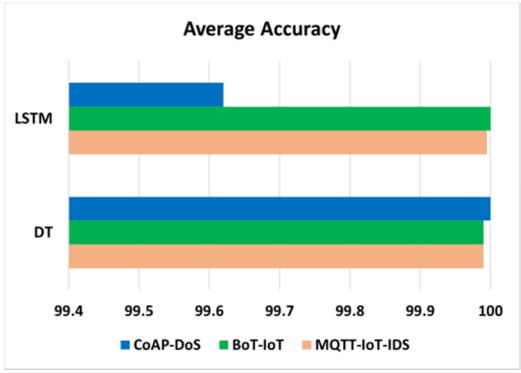

Fig. 2. Accuracy of LSTM and DT for the three datasets.

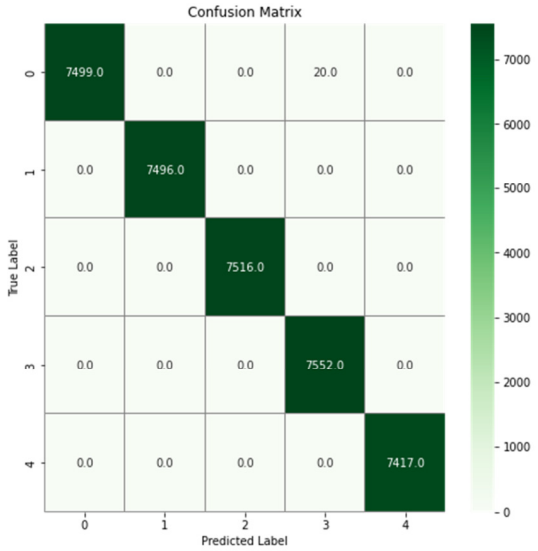

Fig. 3. Confusion matrix of LSTM in MQTT-IoT-IDS.

Another factor that is important in the evaluation of the model is its precision. The precision is calculated as the ratio of positive samples correctly identified to the total number of positive samples classified:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

The recall is calculated as the ratio of true positives to the number of the overall relevant elements:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

The F1-score aims to combine the precision and recall metrics into one metric [7] as follows:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The results shown in Table I confirm the high performance of the DT and LSTM models, which is due to the well-preprocessed data. However, LSTM did not perform well with the CoAP dataset due to the small number of records. Taking this into account, it was deduced that LSTM performs better with large datasets. Additionally, these results indicate that a well-tuned ML model, such as DT, trained on a clean and balanced dataset competes in terms of performance DL models. More importantly, ML models consume fewer resources and require less training and testing time compared to DL. In particular, during the training of the proposed models, DT needed only around 10 s in the training phase using the three considered datasets, while LSTM needed approximately 55 s on the CoAP-DoS. Consequently, the DT model seems to be more suitable for IoT environments where devices are resource-constrained in terms of energy and computation capabilities. Table II shows the average accuracy of many ML and DL models aiming to detect intrusions in IoT environments.

TABLE I. PERFORMANCE OF THE PROPOSED MODELS

	Dataset	Precision	Recall	F1-Score
DT	MQTT-IoT-IDS	99.99	99.99	99.99
LSTM		99.995	99.995	99.995
DT	BoT-IoT	99.99	99.99	99.99
LSTM		100	100	100
DT	CoAP-DDoS	100	100	100
LSTM		67.43	51.41	100

TABLE II. PERFORMANCE COMPARISON OF THE PROPOSED AND EXISTING MODELS ON THE SAME DATASETS

Model	Dataset	Model	Average Accuracy
[24]	MQTT-IoT-IDS	DT	99.92
Proposed DT Model			99.99
[8]		DNN	99.753
Proposed LSTM Model		LSTM	**99.995**
[26]	BoT-IoT	Fine DT	97.43
Proposed DT Model		DT	99.99
[20]		LSTM	99.74
Proposed LSTM Model		**LSTM**	**99.999**
[21]	CoAP-DoS	DT	99.76
Proposed DT model			**100**
[21]		LSTM	99.62
Proposed LSTM model			99.92

For MQTT-IoT-IDS, it is obvious that LSTM reached the highest accuracy of 99.995% compared to the DT models proposed in [19, 26] as well as the proposed DL model. The training and testing times of the proposed method were 136 and 14 s, which are among the average computation times of similar models [8]. Additionally, LSTM outperformed the DNN model proposed in [8]. The proposed LSTM performed better than the models proposed in [15] and Bi-LSTM [27], all trained on the BoT-IoT dataset. However, there is a huge difference between the proposed DT and the models in [28-29], again trained on BoT-IoT. However, DT has the highest accuracy for the CoAP dataset compared to DT and LSTM proposed in [20]. Figure 4 shows the loss function of LSTM for testing and training. The results show that the training and testing accuracies converge with each other, starting from epoch 20 for LSTM-CoAP, epoch 10 for LSTM-MQTT, and approximately epoch 10 for LTSM-BoT. The fluctuation of the loss function for CoAP is due to the small size of the dataset.

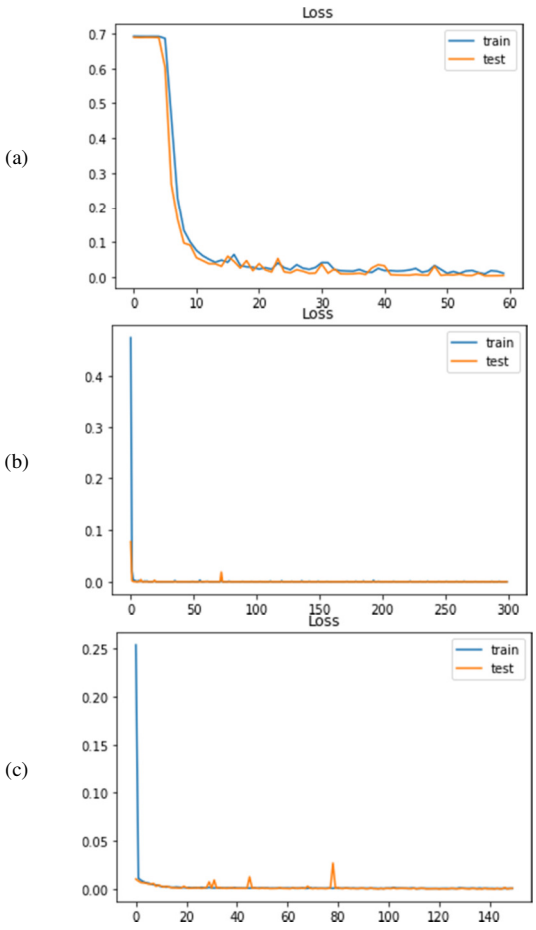

Fig. 4. Loss functions: (a) LSTM-CoAP, (b) LSTM-BoT, and (c) LSTM-MQTT.

C. Deployment of the Models

The following models were deployed in the proposed IP-IDS: LSTM trained on MQTT-IoT-IDS and Bot-IoT datasets, and DT trained on the CoAP-DoS dataset. IP-IDS consists of a three-tier web application: user interface, application server, and database server. At first, the user interface represents a dashboard for the administrator to show the queue of alerts and their status (handled or not), their origin, the timestamp, etc. The application server hosts the proposed ML models and

many other classes responsible for filtering the traffic and classifying it according to the application layer protocol. The database server will be used to store the traffic collected from the network. The Flask web application framework [30] will be used to implement IP-IDS and deploy the ML models, as it provides a panoply of modules that make it easy to develop applications with many details related to protocols and thread management.

IV. CONCLUSION

This study proposed a new IDS for IoT based on ML, called IP-IDS, that can detect MQTT, HTTP, and CoAP-directed intrusions with high accuracy. Existing IDSs for the IoT suffer from many limitations. In particular, the existing models suppose that all IoT network traffic is HTTP, however, new application layer protocols are used today. Additionally, almost all existing models detect intrusions in a single application protocol. Hence, this study proposes a comprehensive IDS that detects intrusions that might threaten three application layer protocols. Furthermore, the Bot-IoT, MQTT-IoT-IDS2020, and CoAP-DDoS IoT-specific datasets were used to train and test the ML models. Two models were considered, DT and LSTM, each trained on the three datasets. The experimental results showed that applying the resample function, which provides balancing datasets and feature selection, on both the MQTT and Bot-IoT datasets allows the enhancement of the detection capabilities of the proposed models and improves their accuracy. The results showed that DT and LSTM in MQTT, Bot-IoT, and CoAP achieved excellent performance in terms of accuracy, F1-score, precision, and recall. The proposed models also outperformed many existing ML/DL models, such as those proposed in [14-15, 27-29]. In future work, this research will focus on investigating the performance of IP-IDS using simulation in an IoT environment.

REFERENCES

[1] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions," *Electronics*, vol. 9, no. 7, Jul. 2020, Art. no. 1177, https://doi.org/10.3390/electronics9071177.

[2] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, Jul. 2019, Art. no. 20, https://doi.org/10.1186/s42400-019-0038-7.

[3] M. Husnain *et al.*, "Preventing MQTT Vulnerabilities Using IoT-Enabled Intrusion Detection System," *Sensors*, vol. 22, no. 2, Jan. 2022, Art. no. 567, https://doi.org/10.3390/s22020567.

[4] G. Nebbione and M. C. Calzarossa, "Security of IoT Application Layer Protocols: Challenges and Findings," *Future Internet*, vol. 12, no. 3, Mar. 2020, Art. no. 55, https://doi.org/10.3390/fi12030055.

[5] K. Aldriwish, "A Deep Learning Approach for Malware and Software Piracy Threat Detection," *Engineering, Technology & Applied Science Research*, vol. 11, no. 6, pp. 7757–7762, Dec. 2021, https://doi.org/10.48084/etasr.4412.

[6] M. H. H. Khairi, S. H. S. Ariffin, N. M. A. Latiff, A. S. Abdullah, and M. K. Hassan, "A Review of Anomaly Detection Techniques and Distributed Denial of Service (DDoS) on Software Defined Network (SDN)," *Engineering, Technology & Applied Science Research*, vol. 8, no. 2, pp. 2724–2730, Apr. 2018, https://doi.org/10.48084/etasr.1840.

[7] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, 2021, Art. no. e4150, https://doi.org/10.1002/ett.4150.

[8] M. A. Khan *et al.*, "A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT," *Sensors*, vol. 21, no. 21, Jan. 2021, Art. no. 7016, https://doi.org/10.3390/s21217016.

[9] T. Gazdar, "An Efficient Intrusion Detection System for Attacks Detection in MQTT Protocol Using Machine Learning," *International Journal of Computer Science and Network Security*, vol. 22, no. 11, pp. 791–798, Nov. 2022, https://doi.org/10.22937/IJCSNS.2022.22.11.110.

[10] E. Jove *et al.*, "Intelligent One-Class Classifiers for the Development of an Intrusion Detection System: The MQTT Case Study," *Electronics*, vol. 11, no. 3, Jan. 2022, Art. no. 422, https://doi.org/10.3390/electronics11030422.

[11] J. Granjal, J. M. Silva, and N. Lourenço, "Intrusion Detection and Prevention in CoAP Wireless Sensor Networks Using Anomaly Detection," *Sensors*, vol. 18, no. 8, Aug. 2018, Art. no. 2445, https://doi.org/10.3390/s18082445.

[12] J. G. Almaraz-Rivera, J. A. Perez-Diaz, and J. A. Cantoral-Ceballos, "Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models," *Sensors*, vol. 22, no. 9, Jan. 2022, Art. no. 3367, https://doi.org/10.3390/s22093367.

[13] B. Susilo and R. F. Sari, "Intrusion Detection in IoT Networks Using Deep Learning Algorithm," *Information*, vol. 11, no. 5, May 2020, Art. no. 279, https://doi.org/10.3390/info11050279.

[14] X. H. Nguyen, X. D. Nguyen, H. H. Huynh, and K. H. Le, "Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways," *Sensors*, vol. 22, no. 2, Jan. 2022, Art. no. 432, https://doi.org/10.3390/s22020432.

[15] T. Gazdar, "A New IDS for Smart Home based on Machine Learning," in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2022, pp. 393–400, https://doi.org/10.1109/CICN56167.2022.10008310.

[16] R. Malik, Y. Singh, Z. A. Sheikh, P. Anand, P. K. Singh, and T. C. Workneh, "An Improved Deep Belief Network IDS on IoT-Based Network for Traffic Systems," *Journal of Advanced Transportation*, vol. 2022, Apr. 2022, Art. no. e7892130, https://doi.org/10.1155/2022/7892130.

[17] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. Den Hartog, "ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485–496, 2021, https://doi.org/10.1109/JIOT.2021.3085194.

[18] M. Anwer, S. M. Khan, M. U. Farooq, and Waseemullah, "Attack Detection in IoT using Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7273–7278, Jun. 2021, https://doi.org/10.48084/etasr.4202.

[19] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset)," in *Selected Papers from the 12th International Networking Conference*, 2021, pp. 73–84, https://doi.org/10.1007/978-3-030-64758-2_6.

[20] J. Mathews, P. Chatterjee, and S. Banik, "CoAP-DoS: An IoT Network Intrusion Data Set," in *2022 6th International Conference on Cryptography, Security and Privacy (CSP)*, Tianjin, China, Jan. 2022, pp. 91–95, https://doi.org/10.1109/CSP55486.2022.00025.

[21] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, Nov. 2019, https://doi.org/10.1016/j.future.2019.05.041.

[22] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *Journal of Big Data*, vol. 8, no. 1, May 2021, Art. no. 65, https://doi.org/10.1186/s40537-021-00448-4.

[23] L. Serrano, *Grokking Machine Learning*. Shelter Island, NY, USA: Simon and Schuster, 2021.

[24] R. G. Kerry *et al.*, "An overview of remote monitoring methods in biodiversity conservation," *Environmental Science and Pollution Research*, vol. 29, no. 53, pp. 80179–80221, Nov. 2022, https://doi.org/10.1007/s11356-022-23242-y.

[25] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 1, pp. 20–28, Mar. 2021, https://doi.org/10.38094/jastt20165.

[26] L. D. Manocchio, S. Layeghy, and M. Portmann, "Network Intrusion Detection System in a Light Bulb," in *2022 32nd International Telecommunication Networks and Applications Conference (ITNAC)*, Wellington, New Zealand, Aug. 2022, pp. 1–8, https://doi.org/10.1109/ITNAC55475.2022.9998371.

[27] K. Saurabh *et al.*, "LBDMIDS: LSTM Based Deep Learning Model for Intrusion Detection Systems for IoT Networks," in *2022 IEEE World AI IoT Congress (AIIoT)*, Jun. 2022, pp. 753–759, https://doi.org/10.1109/AIIoT54504.2022.9817245.

[28] R. A. Manzano Sanchez, M. Zaman, N. Goel, K. Naik, and R. Joshi, "Towards Developing a Robust Intrusion Detection Model Using Hadoop–Spark and Data Augmentation for IoT Networks," *Sensors*, vol. 22, no. 20, Art. no. 7726, Jan. 2022, https://doi.org/10.3390/s22207726.

[29] R. Tekin, O. Yaman, and T. Tuncer, "Decision Tree Based Intrusion Detection Method in the Internet of Things," *International Journal of Innovative Engineering Applications*, vol. 6, no. 1, pp. 17–23, Jun. 2022, https://doi.org/10.46460/ijiea.970383.

[30] "Flask Documentation (3.0.x)." https://flask.palletsprojects.com/en/3.0.x/.