A Network Intrusion Detection System using Deep Learning against MQTT Attacks in IoT

Fatemeh Mosaiyebzadeh, Luis Gustavo Araujo Rodriguez, Daniel Macêdo Batista, R. Hirata Jr.
Department of Computer Science
University of São Paulo (USP), Brazil
{fatemehm, luisgar, batista, hirata}@ime.usp.br

2021 IEEE Latin-American Conference on Communications (LATINCOM) | 978-1-6654-4035-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/LATINCOM53176.2021.9647850

***Abstract*—Cyber-attacks and threats are growing fast in the Internet of Things (IoT) infrastructure as applications in smart cities gain momentum. Usually, IoT devices communicate via machine-to-machine protocols such as Message Queuing Telemetry Transport (MQTT). Due to the heterogeneous structure in IoT and the absence of security by design methodologies, security mechanisms in environments with MQTT traffic are needed, and they can be deployed as Intrusion Detection Systems (IDS). This paper proposes a Deep Learning (DL) based Network IDS trained using a public dataset containing MQTT attacks. We assess the proposal using standard performance metrics such as accuracy, precision, recall, F1-score, and weighted average. When evaluating the performance of our DL-based Network IDS, it obtained, in average, 97.09% of accuracy and an F1-score equal to 98.33% in the detection of MQTT attacks. Another important contribution of our work is the sharing of the experiments on GitHub, which guarantees the reproducibility of the research.**

***Index Terms*—MQTT, IoT, Deep Learning, Cybersecurity.**

I. Introduction

The Internet and the Internet of Things (IoT) platforms improved our lives and interactions, allowing IoT devices to connect remotely across a network infrastructure. Using this infrastructure, data from different domains frequently have been gathered without any human-to-human or human-to-computer interaction [1]. In these environments, publish-subscribe protocols are prevalent, with emphasis to the Message Queuing Telemetry Transport (MQTT) protocol, which offers lightweight-messaging, low-bandwidth consumption, and is the most popular publish-subscribe protocol [2].

The increasing deployment of IoT devices in domains like healthcare, smart cities, and smart homes has brought about principal changes in our daily life [3]. Moreover, our life and society are increasingly overwhelmed with IoT devices. Subsequently, the people and government have been facing unpleasant situations related to IoT in day-to-day life, such as growing cybersecurity attacks and privacy breaches [4].

IoT platforms produce a large amount of valuable and intimate data which needs to be transmitted and analyzed securely. Therefore, the lack of security in IoT applications has shown the need to develop a more efficient and safer environment [5] protected against various security attacks and threats in IoT technologies, such as malware, privacy breaches, Denial of Service (DoS) attacks, security vulnerabilities, and disruption of IoT networks [6].

In recent years, Machine Learning (ML) techniques including supervised learning, unsupervised learning, and reinforcement learning, have been widely used to detect anomalous behavior in the IoT. ML techniques train a model to understand the normal and abnormal actions of the infrastructure or system and detect when a probable new security issue occurs. All of the data generated in the IoT environment can be collected and analyzed to assess the usual interaction pattern between IoT devices. However, not so effective results are obtained by ML techniques in domains with large amount of data. For instance, a pure ML-based Intrusion Detection System (IDS) produces a high rate of false positives when the scenarios are complex in terms of amount of data [7].

To address the complex scenarios and big data in IoT systems, Deep Learning (DL) techniques have become an interesting research topic. DL is a subfield of ML whose advantage, compared to traditional ML, is that it can model big data more simply. Results from the literature have shown that DL methods are fit for this kind of system [8].

In this paper we propose an IDS for IoT networks based on DL techniques to protect against MQTT attacks. To guarantee the reproducibility of our proposal, we trained the DL model, embedded in the IDS, using the public dataset `MQTT-IoT-IDS2020` [9] [10], where IoT devices use the MQTT protocol. The performance evaluation of our DL-based Network IDS shown that it obtained 97.09% of accuracy and an F1-score equal to 98.33% when detecting MQTT attacks.

The rest of this paper is organized as follows: Section II presents related works about IDS for IoT networks. Section III describes the dataset, data preprocessing, and details of the algorithms employed in our proposed IDS. Section IV provides the design of the experiments, carried out to evaluate the performance of the proposed IDS when detecting MQTT attacks, and the obtained results. Section VI presents the conclusions and future work.

II. Related Work

In [11], the authors proposed the usage of two architectures to detect IoT attacks in real-time: Complex Event Processing together with Linear or Support Vector Regression, and ML based on supervised learning. The proposed architectures have been applied to a healthcare IoT network to validate the detection of MQTT attacks.

Ciklabakkal et al. [12] presented an IDS for IoT named ARTEMIS. The ARTEMIS processes data from IoT devices via ML to find the system's normal behavior and generate alerts in case of anomalies. The authors have generated a dataset that contains attacks for MQTT.

Vinayakumar et al. [13] proposed using DL algorithms such as Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN) for intrusion detection in network traffic. The authors evaluated the performance of their proposed DL models over the KDD-Cup 99 dataset, which contains several TCP/IP packets. The experiments depicted that CNN significantly surpasses the performance of classical ML models.

In [14], the authors proposed a DL algorithm to improve the detection of unforeseen cyberattacks on an Internet of Medical Things (IoMT) environment and to establish a reliable and productive IDS. The proposed system, based on Deep Neural Network (DNN), achieved the best performance compared to existing ML proposals. The performance evaluation shows an improvement in accuracy by 15% and a reduction in the computation time by 32%.

III. Methodology

This section describes steps and project decisions taken in the process of proposing a new IDS to detect MQTT attacks in IoT. It is presented the dataset, the data preprocessing, and the details of the algorithms employed in our IDS.

A. Dataset

The dataset employed for the implementation and experimentation of our IDS is known as `MQTT-IoT-IDS2020` and, to the best of our knowledge, it is the only public dataset that addressed unidirectional flows, bidirectional flows, and packet-based features with MQTT traffic [9] [10]. The dataset contains Ethernet traffic gathered by simulation of MQTT sensors. The raw data is captured in `pcap` format, and the authors evaluated features in three levels: unidirectional flow, bidirectional flow, and packet-based features. The dataset is composed of normal instance and four attack scenarios: Aggressive scan, UDP-scan, Sparta SSH brute-force, and MQTT brute-force attack. We focused to detect Aggressive scan, UDP-scan, and mainly MQTT brute-force attack.

In unidirectional flow, the Aggressive scan has 39797 attack instances and 11561 normal instances. The UDP-scan contains 22436 attack samples and 34409 normal samples. MQTT brute-force is composed of 28874 attack instances and 4205 normal instances.

In bidirectional flow, the Aggressive scan has 19907 attack instances and 5786 normal instances. The UDP-scan contains 22434 attack samples and 17230 normal samples. MQTT brute-force is composed of 14544 attack instances and 2152 normal instances.

When evaluating the packet-based features in the dataset, the Aggressive scan has 40624 attack instances and 70768 normal instances. The UDP-scan contains 22436 attack samples and 210819 normal samples. MQTT brute-force is composed of 10013142 attack instances and 1088394 normal instances.

B. Data Preprocessing

The raw data present in the dataset contains some "bad data" because of device errors or human mistakes during the data collection. Therefore, the preprocessing step consisted of identifying and filtering out missing data, duplicate data, and `NULL` values. This step decreases the complexity of the dataset and improves information integrity. Besides, some raw data are imbalanced, and we used the Python imbalanced-learn package [15] for balancing the dataset.

Since the range of some feature values is large, it was needed to utilize the Python Standard scalar function to standardize the dataset into 0 to 1. Also, analyzing and visualizing big data can be challenging. Therefore, to analyze this amount of data with more than 10 million records, we used the Dask library. Also, we used the GridSearchCV function from Scikit-learn to set optimal values for hyperparameters.

C. Algorithms

In this paper, we consider three DL models: a Deep Neural Network (DNN), an LSTM, and a mix of Convolutional and Recurrent Neural Networks (CNN-RNN-LSTM). The overview of our proposed IDS is illustrated in Fig. 1. The first phase of the system is preprocessing of a CSV dataset that has been defined in the Section III-B. The second phase is the training of the deep learning model. In this architecture, a deep learning model has been developed using DNN, CNN-RNN-LSTM, and LSTM. In the training, processed data has been passed to classifiers (DNN, CNN-RNN-LSTM, LSTM). In the last phase, the obtained results demonstrate the performance of our proposed classifier using confusion matrices, accuracy, recall, precision, and F1-score.

A DNN contains simple neurons connected in multiple layers, and it is a popular DL method for complex systems to abstract features and learn as an ML method. An LSTM is an extension of an RNN, which can remember normal and malicious traffic for a long time. An LSTM is widely used due to this property of remembering patterns, which can solve the vanishing gradient problem and learn long-term dependencies.

All algorithms were implemented using TensorFlow and Keras. For evaluating their performance, we compared them with the results obtained by the authors of the `MQTT-IoT-IDS2020` dataset in [10]. In there, the authors evaluated the effectiveness of ML techniques to detect MQTT-based attacks.

The proposed DNN model has an Input layer with 64 neurons. The ReLU activation function follows the layer, and the 0.01 dropout regularization method is also employed. The batch size and learning rate are set to 512 and 0.001. Because the target is multiclass, the softmax activation is used.

The LSTM model contains an input layer and an output layer. In this model, the loss function is categorical Cross-Entropy, and the optimizer is Adam [16]. Also, dropout, epochs, learning rate, and LSTM sizes are 0.1, 100, 0.001, 4, respectively.

The layers of the CNN-RNN-LSTM model consist of an input layer, normalization layers, and an output layer. In the

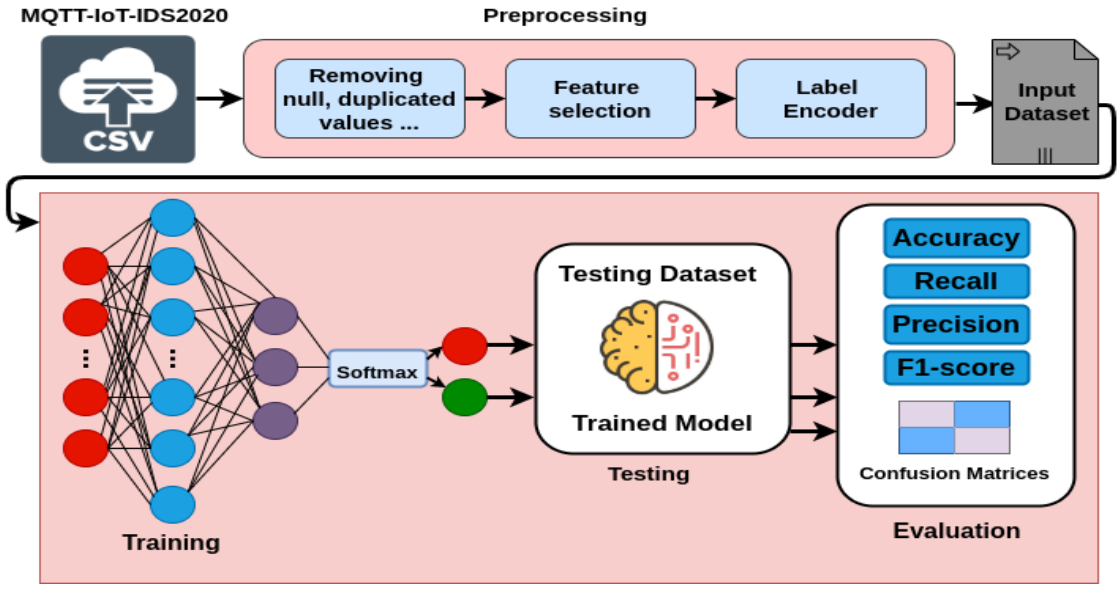

Fig. 1. Deep learning architecture of our proposed models.

proposed CNN-RNN-LSTM scheme, activation functions, loss function, and optimizer are ReLU, categorical-cross-entropy, and Adam, respectively; Whereas, dropout, epochs, and the learning rate are 0.2, 100, and 0.001. Moreover, in this model, the LSTM-size is equal to 20.

IV. Experimental Design

In the process of designing a new IDS, it is important to evaluate the best DL model to be deployed. This section describes the performance evaluation of the three DL models presented in Subsection III-C. Software and hardware resources used are detailed and the evaluation metrics are explained. Also, it discusses the results and compares the performance of our proposed DL models with the one obtained by the Decision Trees (DT) technique implemented in [10]. This technique was the one which obtained the best overall performance among six different ML techniques when detecting attacks present in the `MQTT-IoT-IDS2020` dataset by the authors of the dataset. The results are separated in three groups, related to the three abstraction levels of features presented in the dataset: bidirectional flow, unidirectional flow, and packet-based. To allow the reproduction of all the experiments executed and reported in this section, we shared them in the Python notebook at https://github.com/fatemehm/IDS-DeepLearning.

A. Experimental Setup

For the experimental setup, we worked on the Google Colaboratory. The software used to implement the algorithms was Python 3, Dask, Pandas, Numpy, Keras, Tensorflow, and Scikit-learn.

B. Results of Bidirectional Flow

Table I shows the confusion matrix of our proposed models for the bidirectional flow. It can be seen that in the Aggressive scan detection, the DNN model has achieved zero False Positives (FP) and False Negatives (FN) in intrusion detection, but in other types of attack, the model did not obtain the same performance. It obtained 4428 True Positives (TP) and 15 FP in UDP scan. In the MQTT brute-force attack, the DNN model obtained zero FN, and 3533 TP, but 111 FP. When we concatenated and balanced the entire attack samples, our proposed DNN model obtained 22115 TP and 238 FP.

TABLE I
Results of TP, TN, FP, and FN for DNN, CNN-RNN-LSTM, and LSTM algorithm - Bidirectional flow

Algorithm	Attacks	TP	TN	FP	FN
DNN	Aggressive scan	4924	5030	0	0
	UDP scan	4428	4525	15	6
	MQTT brute-force	3533	3628	111	0
	Overall Bidirectional	22115	22116	238	2
CNN-RNN-LSTM	Aggressive scan	4924	5030	0	0
	UDP scan	5463	5698	56	0
	MQTT brute-force	3533	3628	111	0
	Overall Bidirectional	22088	22116	265	2
LSTM	Aggressive scan	4924	5029	0	1
	UDP scan	5469	5692	50	6
	MQTT brute-force	3533	3626	111	2
	Overall Bidirectional	22270	22065	83	53

Our proposed CNN-RNN-LSTM model for the bidirectional flow, in Aggressive scan, achieved zero FP and zero FN. Also, it obtained zero FN in the UDP scan, but it classified 5463 TP and 56 FP. In the MQTT brute-force attack, it classified 3533 TP and 111 FP. After concatenating all attacks instances and balancing the data, CNN-RNN-LSTM obtained 22088 TP and just 265 FP.

LSTM model, in Aggressive scan detection, obtained zero FP and 1 FN. In UDP scan, LSTM achieved 6 FN, 5469 TP, and just 50 FP. Also, in the MQTT brute-force attack, LSTM classified 3533 TP and 111 FP. Finally, by concatenating all attacks samples and balancing data, LSTM obtained 22270 TP and 83 FP.

TABLE II
RESULTS OF ACCURACY, PRECISION, RECALL, F1-SCORE AND WEIGHTED AVERAGE FOR DNN, CNN-RNN-LSTM, AND LSTM COMPARED TO THE BASELINE (DT) - BIDIRECTIONAL FLOW

Algorithm	Attacks	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	Weighted-avg(%)
DNN	Aggressive scan	100	100	100	100	100
	UDP scan	99.77	100	100	100	100
	MQTT brute-force	98.47	100	97.0	98.0	99.0
	Overall Bidirectional	99.59	100	99.0	99.0	100
CNN-RNN-LSTM	Aggressive scan	100	100	100	100	100
	UDP scan	99.50	100	99.0	99.0	100
	MQTT brute-force	98.47	100	97.0	98.0	99.0
	Overall Bidirectional	99.64	100	99.0	99.0	100
LSTM	Aggressive scan	99.99	100	100	100	100
	UDP scan	99.50	100	99.0	99.0	100
	MQTT brute-force	98.45	100	97.0	98.0	98.0
	Overall Bidirectional	99.64	100	99.0	99.0	100
DT [10]	Aggressive scan	-	99.9	100	99.95	99.95
	UDP scan	-	100	100	100	99.95
	MQTT brute-force	-	99.93	99.93	99.93	99.95
	Overall Bidirectional	99.95	-	-	-	-

Table II compares all the three proposed models with the DT model, used as baseline. In Aggressive scan attacks, DNN and CNN-RNN-LSTM models obtained accuracy and F1-score equal to 100%. In UDP scan attacks, the DNN model achieved the best result with an accuracy equal to 99.77% and an F1-score of 100%. In MQTT brute-force, both DNN and CNN-RNN-LSTM models obtained an accuracy of 98.47% and an F1-score of 98%. Overall, the LSTM and CNN-RNN-LSTM models obtained accuracy equal to 99.64% and F1-score of 99%, while the DNN algorithm obtained 99.59% of accuracy and 99% of F1-score. Comparing the performance of our proposed models with the baseline, we note that the best accuracy of our models was 99.64%, against 99.95% of the baseline. However, in general, our models achieved best results in relation to the other metrics. It is important to explain that not all the metrics are present in [10]. This is why some cells in the table are filled with "-".

C. Results of Unidirectional flow

Table III shows the confusion matrix of the DNN model for the unidirectional flow. Our proposed DNN model obtained zero FP and zero FN when classifying traffic in the presence of Aggressive scan. The DNN model obtained zero FN in the UDP scan classification, but it classified 72 instances of attack as normal traffic. In the MQTT brute-force attack, DNN classified zero FN, but it considered 113 samples of attack as normal traffic. By concatenating and balancing the entire attacks, DNN achieved 44076 TP and 422 FP.

In the classification of Aggressive scans, the CNN-RNN-LSTM achieved 11 FP and zero FN. The performance of CNN-RNN-LSTM in UDP scan classification shows that it achieved zero FN and 70 FP. In MQTT brute-force, the CNN-RNN-LSTM also obtained zero FN, but 113 FP. After balancing and concatenating attacks, CNN-RNN-LSTM classified 44208 TP and 290 FP.

LSTM model for unidirectional flow, in Aggressive scan classifications, achieved zero FP and zero FN . In the UDP scan attack, the LSTM classified zero FN and 73 FP. The LSTM obtained 5 FN and 112 FP in MQTT brute-force. By balancing and concatenating attacks, LSTM classified 44278 TP and 220 FP.

TABLE III
RESULTS OF TP, TN, FP, AND FN FOR DNN, CNN-RNN-LSTM, AND LSTM ALGORITHM - UNIDIRECTIONAL FLOW

Algorithm	Attacks	TP	TN	FP	FN
DNN	Aggressive scan	7927	7992	0	0
	UDP scan	6750	6942	72	0
	MQTT brute-force	5598	5839	113	0
	Overall Bidirectional	44076	44299	422	0
CNN-RNN-LSTM	Aggressive scan	9944	9944	11	0
	UDP scan	6752	6942	70	0
	MQTT brute-force	5598	5839	113	0
	Overall Bidirectional	44208	44233	290	74
LSTM	Aggressive scan	9955	9944	0	0
	UDP scan	6749	6942	73	0
	MQTT brute-force	5599	5834	112	5
	Overall Bidirectional	44278	44244	220	63

Table IV shows the performance of our proposed models and compares them with the DT model from the literature. Similar to the results obtained with the features from bidirectional flow, the baseline obtained best accuracy. However, when evaluating the other metrics, specifically for the classification of MQTT brute-force, results were not too distant. In particular, when evaluating the precision, all the proposed models were better than the baseline.

D. Result of Packet-Based

Table V shows the confusion matrix of the our proposed models for classification considering packet-based features. It can be seen that in the Aggressive scan classification and in the UDP scan classification, our DNN model had achieved zero FP and zero FN. In the MQTT brute-force attack, it obtained 2003879 TP and just 505 FP. Nonetheless, with balanced and concatenated data, the DNN model has achieved 2015320 TP and 772 FP.

CNN-RNN-LSTM model for packet-based features, in Aggressive scan and UDP scan classifications, achieved zero FP and zero FN . In the MQTT brute-force attack, the CNN-RNN-LSTM model achieved zero FN, 2003879 TP, and 505 FP. In the end, after concatenated entire attacks and balanced the data set, the model obtained 2015475 TP and 617 FP.

TABLE IV
RESULTS OF ACCURACY, PRECISION, RECALL, F1-SCORE AND WEIGHTED AVERAGE FOR DNN, CNN-RNN, AND LSTM COMPARED TO THE BASELINE (DT) - UNIDIRECTIONAL FLOW

Algorithm	Attacks	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	Weighted-avg(%)
DNN	Aggressive scan	100	100	100	100	100
	UDP scan	99.48	100	99.0	99.0	99.0
	MQTT brute-force	99.02	100	98.0	99.0	99.0
	Overall Unidirectional	99.52	100	99.0	100	100
CNN-RNN-LSTM	Aggressive scan	99.94	100	100	100	100
	UDP scan	99.49	100	99.0	99.0	99.0
	MQTT brute-force	99.02	100	98.0	99.0	99.0
	Overall Unidirectional	99.59	100	99.0	100	100
LSTM	Aggressive scan	100	100	100	100	100
	UDP scan	99.47	100	99.0	99.0	99.0
	MQTT brute-force	98.99	100	98.0	99.0	99.0
	Overall Unidirectional	99.68	100	99.0	100	100
DT [10]	Aggressive scan	-	99.95	100	99.97	99.96
	UDP scan	-	100	99.91	99.96	99.96
	MQTT brute-force	-	99.95	99.95	99.95	99.96
	Overall Unidirectional	99.96	-	-	-	-

In an Aggressive scan attack, the LSTM achieved zero FP and zero FN, also in UDP scan and MQTT brute-force attacks, LSTM obtained zero FN. However, in the UDP scan, it achieved 52,478 TP and just 3 FP. In MQTT brute-force attack, LSTM gained 2,003,879 true-positive and 505 false-positive. Lastly, with the concatenated data, LSTM obtained 2,015,554 TP and 538 FP.

TABLE V
RESULTS OF TP, TN, FP, AND FN FOR DNN, CNN-RNN-LSTM, AND LSTM ALGORITHM - PACKET-BASED

Algorithm	Attacks	TP	TN	FP	FN
DNN	Aggressive scan	17649	17735	0	0
	UDP scan	52481	52929	0	0
	MQTT brute-force	2003879	1849306	505	151567
	Overall Packet-based	2015320	89271	772	183874
CNN-RNN-LSTM	Aggressive scan	17649	17735	0	0
	UDP scan	52481	52929	0	0
	MQTT brute-force	2003879	2000873	505	0
	Overall Packet-based	2015475	91525	617	181620
LSTM	Aggressive scan	17649	17735	0	0
	UDP scan	52478	52929	3	0
	MQTT brute-force	2003879	2000873	505	0
	Overall Packet-based	2015554	88662	538	184483

Table VI shows the performance of the models. Now, it is observed that in all classifications and considering any metric, the proposed models are better than the baseline. Among the three proposed models, the CNN-RNN-LSTM obtained the best result and should be the model implemented in a Network IDS based on DL to detect MQTT attacks in IoT. Also, features at the level of packet are the best choice to have the best detection.

V. DISCUSSION

In this section, we will discuss the results of our proposed deep learning strategy. Also, the results of our models are compared to the previous results of another related work.

The results of bidirectional flow shows, in the aggressive scan attack, that our models were more significantly effective at identifying attacks. In the UDP scan, DNN shows 15 FP, which is less than the FP of CNN-RNN-LSTM and LSTM. In the MQTT brute-force attack, the performance of our proposed models, at identifying attacks, was closer to each other. Moreover, Table II shows the detection rate of our proposed models in the bidirectional flow. Overall, CNN-RNN-LSTM and LSTM achieved an accuracy of 99.64% and an F1-score of 99% for normal and attack classes. We note that the accuracy of the baseline was 99.95%, which is better than our models in the bidirectional flow.

The results of unidirectional flow shows, in the aggressive scan attack, that DNN and LSTM were more significantly effective at identifying attacks. In the UDP scan and MQTT brute-force attack, the performance of our proposed models at identifying attacks was closer to each other. Furthermore, the LSTM algorithm, when compared to our other proposed models, shows the best performance. The detection rate of our proposed models shows that, in the unidirectional flow, LSTM achieved an accuracy of 99.68% and an F1-score of 100% for normal and attack classes. We note that the accuracy of the baseline was 99.96%, which is better than our models in the unidirectional flow.

The results of packet-based shows that in the aggressive scan and UDP scan attacks, all of our proposed models were significantly effective at identifying attacks. In the MQTT brute-force attack, our proposed models show 505 FP and zero FN. It can be seen that LSTM, in all of the three levels, was more effective at identifying attacks when compared to DNN and CNN-RNN-LSTM.

Moreover, Table VI shows the detection rate of our proposed models in the packet-based. In unidirectional and bidirectional flow, comparing the performance of our proposed models with the baseline, we note that the baseline obtained the best accuracy. However, in the packet-based, our proposed models show the best performance. In the aggressive scan and UDP scan attacks, all of our proposed models achieved an accuracy of 100%, and an F1-score of 100% for normal and attack classes. In MQTT brute-force attacks, CNN-RNN-LSTM and LSTM achieved an accuracy of 99.99% and an F1-score of 100% for normal and attack classes. Comparing the performance of CNN-RNN-LSTM and LSTM models, in MQTT brute-force attacks, with the baseline we note that the F1-score of our models was 100%, against 72.51% of the

TABLE VI
RESULTS OF ACCURACY, PRECISION, RECALL, F1-SCORE AND WEIGHTED AVERAGE FOR DNN, CNN-RNN-LSTM, AND LSTM COMPARED TO THE BASELINE (DT) - PACKET-BASED

Algorithm	Attacks	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	Weighted-avg(%)
DNN	Aggressive scan	100	100	100	100	100
	UDP scan	100	100	100	100	100
	MQTT brute-force	96.20	93.0	100	96.0	96.0
	Overall Packet-based	91.93	92.0	100	96.0	93.0
CNN-RNN-LSTM	Aggressive scan	100	100	100	100	100
	UDP scan	100	100	100	100	100
	MQTT brute-force	99.99	100	100	100	100
	Overall Packet-based	92.04	92.0	100	96.0	93.0
LSTM	Aggressive scan	100	100	100	100	100
	UDP scan	100	100	100	100	100
	MQTT brute-force	99.99	100	100	100	100
	Overall Packet-based	91.92	92.0	100	96.0	93.0
DT [10]	Aggressive scan	-	99.98	100	99.99	88.55
	UDP scan	-	100	99.98	99.99	88.55
	MQTT brute-force	-	72.47	72.56	72.51	88.55
	Overall Packet-based	88.55	-	-	-	-

baseline. Overall, CNN-RNN-LSTM achieved an accuracy of 92.04% and an F1-score of 96% for normal and attack classes. Comparing the performance of CNN-RNN-LSTM model with the baseline, we note that the accuracy of CNN-RNN-LSTM was 92.04%, against 88.55% of the baseline.

VI. CONCLUSIONS AND FUTURE WORK

This paper proposed and evaluated DL models for IoT intrusion detection in scenarios with MQTT traffic. We used `MQTT-IoT-IDS2020` dataset to validate our proposed DNN, CNN-RNN-LSTM and LSTM models. The results obtained shows that CNN-RNN-LSTM and LSTM models have a very close performance. On average, the overall result of CNN-RNN-LSTM is slightly better than all other models, with an accuracy of 97.09% and an F1-score of 98.33%. Our future work intends to implement a Network IDS based on the CNN-RNN-LSTM model and deploy it in a testbed to evaluate its performance in real-time attack detection.

ACKNOWLEDGMENTS

This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, FAPESP proc. 14/50937-1, and FAPESP proc. 15/24485-9. It is also part of the FAPESP proc. 18/22979-2 and FAPESP proc. 18/23098-0.

REFERENCES

[1] K. Fizza, A. Banerjee, K. Mitra, P. P. Jayaraman, R. Ranjan, P. Patel, and D. Georgakopoulos, "QoE in IoT: A Vision, Survey and Future Directions," *Discover Internet of Things*, vol. 1, no. 1, pp. 1–14, 2021.

[2] L. G. Araujo Rodriguez and D. Macêdo Batista, "Program-Aware Fuzzing for MQTT Applications," in *Proceedings of the 29th ACM SIGSOFT ISSTA*, 2020, p. 582–586.

[3] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. Leung, "Enabling Massive IoT Toward 6G: A Comprehensive Survey," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11 891–11 915, 2021.

[4] G. Arbex, K. Machado, M. N. Lima, D. M. Batista, and R. Hirata Jr., "IoT DDoS Detection Based on Stream Learning," in *Proceedings of the 12th International Conference on Network of the Future (NoF*, 2021.

[5] H. Kwon, J. E. Fischer, M. Flintham, and J. Colley, "The Connected Shower: Studying Intimate Data in Everyday Life," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–22, 2018.

[6] J. Deogirikar and A. Vidhate, "Security Attacks in IoT: A Survey," in *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2017, pp. 32–37.

[7] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine Learning on Big Data: Opportunities and Challenges," *Neurocomputing*, vol. 237, pp. 350–361, 2017.

[8] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[9] H. Hindy, C. Tachtatzis, R. Atkinson, E. Bayne, and X. Bellekens, "MQTT-IoT-IDS2020: MQTT Internet of Things Intrusion Detection Dataset," 2020. [Online]. Available: https://dx.doi.org/10.21227/bhxy-ep04

[10] 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.

[11] J. Roldán, J. Boubeta-Puig, J. L. Martínez, and G. Ortiz, "Integrating Complex Event Processing and Machine Learning: An Intelligent Architecture for Detecting IoT Security Attacks," *Expert Systems with Applications*, vol. 149, p. 113251, 2020.

[12] E. Ciklabakkal, A. Donmez, M. Erdemir, E. Suren, M. K. Yilmaz, and P. Angin, "ARTEMIS: An Intrusion Detection System for MQTT Attacks in Internet of Things," in *Proceedings of the 38th Symposium on Reliable Distributed Systems (SRDS)*, 2019, pp. 369–3692.

[13] R. Vinayakumar, K. Soman, and P. Poornachandran, "Applying Convolutional Neural Network for Network Intrusion Detection," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1222–1228.

[14] S. P. RM, P. K. R. Maddikunta, M. Parimala, S. Koppu, T. R. Gadekallu, C. L. Chowdhary, and M. Alazab, "An Effective Feature Engineering for DNN using Hybrid PCA-GWO for Intrusion Detection in IoMT Architecture," *Computer Communications*, vol. 160, pp. 139–149, 2020.

[15] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning," *Journal of Machine Learning Research*, vol. 18, pp. 1–5, 2017.

[16] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.