A Hybrid CNN-LSTM Model for IIoT Edge Privacy-Aware Intrusion Detection

Erik Miguel de Elias*, Vinicius Sanches Carriel*, Guilherme Werneck de Oliveira*,
Aldri Luiz dos Santos†, Michele Nogueira†, Roberto Hirata Junior*, Daniel Macêdo Batista*
*Department of Computer Science
University of São Paulo (USP), Brazil
edeelias@acm.org, {vcarriel, werneck, hirata, batista}@ime.usp.br
†Department of Computer Science
Federal University of Minas Gerais (UFMG), Brazil
{aldri, michele}@dcc.ufmg.br

2022 IEEE Latin-American Conference on Communications (LATINCOM) | 978-1-6654-8225-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/LATINCOM56090.2022.10000468

***Abstract*—Security is a critical issue in the context of IoT and, more recently, of Industrial IoT (IIoT) environments. To mitigate security threats, Intrusion Detection Systems have been proposed. Still, most of them can achieve high accuracy only by having access to the application layer of the flows, which is problematic in terms of privacy. This paper presents a neural network model based on a hybrid CNN-LSTM architecture to detect several attacks in the network traffic at the Edge of IIoT using only features from the transport and network layers. Besides improving privacy, the proposal achieves 97.85% average accuracy when classifying the traffic as benign or malicious and 97.14% average accuracy when classifying 15 specific attacks in a dataset containing IIoT traffic. Moreover, all the code produced is available as free software, facilitating new studies and the reproduction of the experiments.**

***Index Terms*—IoT, IIoT, Neural Networks, Deep Learning, Machine Learning, Intrusion Detection**

I. Introduction

The advantages of wireless communications are boosting the usage of IoT devices, which have low processing power architecture and low energy to stay isolated and share data in real-time. As reported in [1], security and privacy are always a primary concern in any application, and IoT environments demand a proper security check at regular intervals, requiring customized security and privacy levels. Reasonable technology, law, and regulation efforts are found around the IoT security theme.

Industrial IoT (IIoT) is a subset of the IoT, sharing attributes related to connectivity, security, energy, and computation limitations. IIoT systems use more accurate devices to attend to the automation and production in the industrial process. Due to its value and impact on power and influence in our society, industrial sites are targets of malicious threats [2]. Network traffic is a natural topic of discussion when one has security in mind. Since deep packet inspection is costly, and IIoT devices do not have enough processing, energy, and storage capacities to implement it, identifying attacks at the network's edge must take alternative solutions as deep learning techniques for traffic classification [3]. The traffic payload analysis and features carrying personal data such as IP address, browser details, device information, and many other data found in the application layer endorse privacy concerns discussion.

Intrusion detection systems spot attacks nowadays with accuracies near 100% [2], [4] thanks to the use of modern machine learning techniques. For instance, Ferrag et al. [5] apply traditional machine learning algorithms and deep learning neural networks to classify traffic in a testbed with IoT and IIoT devices and applications under six types of attacks (the traffic captured in this testbed was used to create the Edge-IIoTset dataset [6]). Despite the good results reported in [5], the authors used only a small part of the entire dataset to train and validate the model. Besides that, some of the features used may have greater importance for a particular type of attack, which would help the model identify more easily specific attack classes. For example, an HTTP flood DDoS attack can be easily identified if one knows that the packets carry HTTP messages. Moreover, by analyzing the dataset is also possible to find out that some specific values related to MQTT sensors are not present on attack features, showing extreme correlated features to normal traffic and explaining the success of 100% (for two-class) and around 94% (for 15 or five classes). This evidence suggests that it is not difficult to identify some attacks if the application layer features are considered, leading to an if-else solution rather cheaper than deep learning. Additionally, the results may not be extendable to all the dataset traffic since, from all ten IoT devices of the testbed, only one of them was considered by the authors.

Inspired by the previous analysis, this paper poses the following research question: **is it possible to classify, with high accuracy, the benign and malicious network traffic without specific features from the application layer in an IIoT environment?** To answer the question, we trained a deep learning architecture with the Edge-IIoTset, using features of the TCP, UDP, ICMP, and ARP protocol, ignoring the HTTP, DNS, MQTT, and ModBus protocols features. This approach favors privacy, processing capability, and data manipulation. Moreover, we expect that the chosen features be generic but enough for traffic classification without deep packet inspection. We found five core advantages of applying this strategy during this research. Besides improving privacy, the

approach achieves 97.85% average accuracy when classifying the traffic as benign or malicious and 97.14% average accuracy when classifying 15 specific attacks in a dataset containing IIoT traffic.

This work contributes with: (1) a clear understanding of the advantages, in terms of security, of the proposed approach for traffic classification by removing the application layer features; (2) a hybrid CNN (Convolutional Neural Network)-LSTM (Long Short-Term Memory) neural network model that can be used in an intrusion detection system favoring data privacy; (3) a more realistic analysis of the network traffic from the Edge-IIoTset dataset, including all devices present on the testbed, and using a more significant part of the dataset; (4) advancements over some previous studies ([7], and [5]) with new analyses and insight; (5) providing the code used in this research as free software for public use, facilitating new studies and this study comprehension.

The next sections are organized as it follows: Section II explores the literature about IoT, IIoT, and traffic classification; Section III presents the proposed model; Section IV explains the experimental design; Section V presents the achieved results compared with other works; and Section VI presents our final thoughts and possible future studies.

II. Related Work

Despite this work focusing on signature-based (also known as misuse-based) attack detection, it is sightly to point to two other types, the anomaly-based and hybrid approaches. The signature type uses known attacks based on specific traffic attributes and is tipically effective with a low frequency of false alarms. Hence, it has to be constantly updated with new traffic variance, and attacks [8]. On the other hand, the anomaly-based type mock-ups the legitimate traffic behavior and indicates anything also as an attack. The anomaly-based technique has the advantage of warning about unknown attacks but has the possibility of high false alarm rates [8]. The hybrid approach combines previous techniques to minimize the false alarm and increase the detection capability of new attacks.

According to [9] and [8], there are many studies for Intrusion Detection Systems focused on the IoT field. However, the datasets do not fit in a real-world IoT scenario, lacking in mimicking reality. Furthermore, most datasets usually used in the literature are pre-processed, meaning that the features are processed according to a strategy not representing the exact feature from the network data frame. So, it needs more pre-processing steps for classification, making real-time implementations difficult.

The possibility of using different datasets (see Table I) for this research was considered. However, most of them are flow-preprocessed and it is not possible to distinguish clearly the data from the application layer. Most datasets contain IIoT traffic, but, except for the Edge-IIoTset dataset, when the application layer features are removed, there is a significant loss of information. For instance, after dropping the features of the MQTTset there will be only three features for the analysis in the TCP layer, and in the case of MQTT-IOT-IDS2020 the TCP features are control flags features.

TABLE I: Cyber security IoT datasets investigated

Dataset Name	Year	IIoT
CICIDS2017 [10]	2017	No
MQTT-IOT-IDS2020 [11]	2020	Yes
MQTTset [12]	2020	No
WUSTL-IIoT-2021 [13]	2021	Yes
X-IIoTID [14]	2021	Yes
Edge-IIoTset [6]	2022	Yes

Nonetheless, the scope of this study is not to create, compound, or treat datasets as other works like [15], which combine different datasets to build a new one to be analyzed. In their study, five datasets (BoT-IoT, IoT-NI, MQTT-IoT-IDS2020, MQTTset, and IoT-23) were rebuilt from the PCAP files using the CICFlowMeter [16] tool. The resulting dataset is called IoT-DS2.

To the best of our knowledge, no study using datasets to classify and detect attacks in an IoT or IIoT network excluding the application layer has been proposed before. In this context, studies that use the Edge-IIoTset dataset, such as [5] and [17], are the most similar to our work.

In [5] a testbed was implemented involving cloud, NFV, SDN, and Edge layers, and various IoT and IIoT devices subjected to normal or legitimate, and attack traffic, generating a vast dataset allowing deep learning studies. The authors used the testbed to generate normal traffic and five attack types: DoS/DDoS, Man in the Middle (MITM), information gathering, injection, and Malware. The authors exemplify the usage of the dataset by providing a sample file (named Selected ML Dataset) to evaluate machine learning models. Two approaches were considered: centralized and federated learning. For centralized learning, the Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), as well as a Deep Neural Network (DNN) were used, while for federated learning, the same DNN of the centralized was used. However, only a limited part of the traffic is analyzed. Despite the good result (above 94% accuracy for 15 multiclass and 99.99% for binary classification with centralized learning), it is not extensible to the entire dataset. The Selected ML Dataset for ML analysis covers, on average, 36% of the attack records, and only the DHT11 device sensor of temperature and humidity from all ten IoT-based devices, which is only 1.49% of the DHT11 records. The analysis covers 0.22% of all dataset normal traffic records. In general, their analyses cover only 0.75% of all dataset. In our study, all devices on the testbed are used, and a more significant part of the dataset is considered.

In [17] a detection approach aimed at the healthcare industry was proposed. It employed an optimized LightGBM (Light Gradient Boosting Machine) model and a BERT (Bidirectional Encoder Representations)-based Transformer model. Beyond the Edge-IIoTset, three other datasets were evaluated. Regarding the Edge-IIoTset, the same part of the Selected ML Dataset used in [5] was used, which means that the diversity of sensors

and the representativeness of the testbed are compromised as in [5]. The authors reached over a 99% score using the ROC-AUC metric when classifying the four datasets. For binary classification, the LightGBM model got a perfect 100% accuracy, and for 15 multiclass classifications 92% precision, 88% Recall, and 89% F1-score on average for 14 types of attack besides the normal class.

Roopak et al. [7] propose a hybrid CNN-LSTM model for detecting cyberattacks in IoT. The proposed model uses features from the application layer, and the authors evaluate their proposal using the CICIDS2017 dataset for DDoS attack detection achieving an accuracy of 97.16%.

III. The Hybrid CNN-LSTM Model

When applied to classify network traffic, deep learning neural networks have achieved good results. The usual pipeline when working with such methods is firstly to implement or use an open-source implementation to test a certain hypothesis and try to modify the architecture if the proposed model does not perform well.

The hybrid model proposed by Roopak et al. [7] is a CNN and LSTM implementation beginning with a 1D convolution layer followed by an LSTM layer, a dropout layer, a fully connected layer, ending with a dense layer activated by a `sigmoid` function.

To improve the model's performance when dropping the application layer information, we propose modifying the original architecture and adding two more dense layers and a flattened layer before the last dense layer. The proposed modification has been made by empirical decision of the authors, as already discussed on the literature that increasing the layers improves the models metrics [18].Fig. 1 illustrates the implemented architecture presenting the specific parameters for each layer. Since the Conv1d layer needs a 3D shape (like [batch, step, channels]), the original dataset was transformed using a `reshape` function to be [18,1] shape. The Conv1d layer uses five as a value for kernel size and the LSTM layer uses 128 as units with the `tanh` activation function. The `relu` activation is used for the Conv1d and Dense layers, while the last layer applies the `softmax` activation function for multiclass output or `sigmoid` for binary classification.

We are considering that this model would be implemented at the edge of an IIoT infrastructure by receiving traffic both from IoT and IIoT devices.

IV. Experimental Design

The dataset used to evaluate the proposed model is the Edge-IIoTset [6]. It was chosen due to its realistic testbed for IoT and IIoT devices. The classes of the traffic present in the dataset are: Normal traffic (Normal), Backdoor attack (Back), HTTP flood DDoS attack (HTTP), ICMP flood DDoS attack (ICMP), TCP SYN Flood DDoS attack (TCP), UDP flood DDoS attack (UDP), OS Fingerprinting attack (Fing), Man in the middle attack (MITM), Password cracking attack (Pwd), Port Scanning attack (Port), Ransomware attack (Rans),

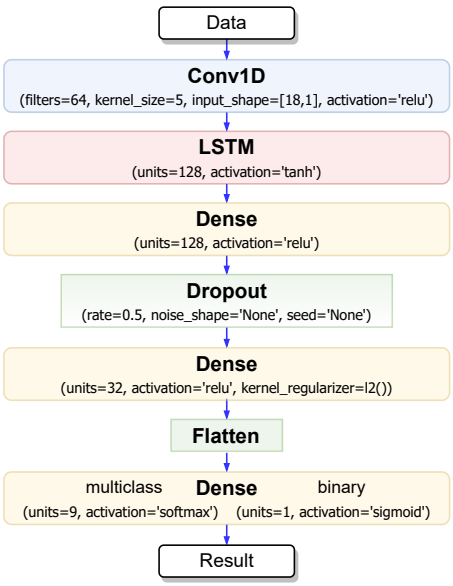

Fig. 1: Proposed Hybrid CNN-LSTM Neural Network Model.

SQL Injection (SQL), Upload attack (Upload), Vulnerability scanning attack (Scan), and Cross-site Scripting attack (XSS).

The dataset is provided as PCAP and CSV files distributed into folders according to their types (normal traffic and attack traffic). Inside the attack folder, there are files specific to the attacks and inside the normal folder, there are files specific to the sensors. In this study, the traffic of all sensors has been used. However, the Modbus traffic was not considered because when pre-analyzing the data much information specific to this protocol was broken and missing.

The application layer features from the HTTP, DNS, and MQTT protocols were removed. Even after removing the application layer features, there is still useful information at the transport and network layers which we claim to be enough for good classification, which also increases the security since the attacker would have to mimic the normal data traffic flow much better to avoid detection. Also, features like port numbers and IP addresses were excluded based on their high correlation to traffic type. In the end, 18 features are used for classification from all 63 original features. In summary, the "cleaning" of the dataset follows these steps:

1) To read each CSV file from the respective folder;
2) To drop rows with IP address, not in IoT device list;
3) To drop unwanted features;
4) To convert features from hex to float;
5) To drop rows with IP address values at *arp.hw.size* and *arp.opcode* features;
6) To drop duplicated rows and with null value;
7) To reserve random sample for test and train.

The cleaning process resulted in a processed dataset with 26% fewer records than the original, and it was divided into nine subsets, one for testing and eight for training to allow segmented analysis and observations of the metrics as the records increment in the training set. The testing subset has

TABLE II: Dataset preprocessed and subsets statistics (number of records)

Type	Class	Original	Test (20%)	Train (80%)	Train (60.0%)	Train (40.0%)	Train (20.0%)	Train (10.0%)	Train (5.0%)	Train (2.6%)	Train (0.9%)
Attack Traffic	Back	24862	4805	19221	14416	9611	4806	2403	2000	2000	2000
	HTTP	229022	41206	164823	123617	82411	41206	20603	10302	5151	2000
	ICMP	2914354	26215	104859	78644	52429	26214	13107	6554	3277	2000
	TCP	2020120	403788	1615151	1211363	807575	403788	201894	100947	50474	16152
	UDP	3201626	640220	2560882	1920662	1280441	640220	320110	160055	80028	25609
	MITM	1229	72	288	288	288	288	288	288	288	288
	Fing	1001	171	682	682	682	682	682	682	682	682
	Pwd	1053385	200627	802507	601880	401253	200626	100313	50156	25078	8025
	Port	22564	3995	15982	11986	7991	3996	2000	2000	2000	2000
	Rans	10925	1938	7751	5813	3875	2000	2000	2000	2000	2000
	SQL	51203	10165	40661	30496	20331	10166	5083	2542	2000	2000
	Upload	37634	7361	29446	22084	14723	7362	3681	2000	2000	2000
	Scan	145869	29108	116433	87325	58217	29108	14554	7277	3638	2000
	XSS	15915	3013	12053	9040	6027	3014	2000	2000	2000	2000
	Subtotal	**9729709**	**1372684**	**5490739**	**4118296**	**2745854**	**1373476**	**688718**	**348803**	**180616**	**68756**
Normal Traffic	Distance	1143540	184018	736074	552056	368037	184018	92009	46004	23002	7361
	Flame	1070196	171428	685712	514284	342856	171428	85714	42857	21428	6857
	H. Rate	165319	26738	106950	80212	53475	26738	13369	6684	3342	2000
	IR	1307778	209299	837197	627898	418599	209300	104650	52325	26162	8372
	phValue	746908	115183	460731	345548	230365	115182	57591	28796	14398	4607
	Soil M.	1192777	180562	722250	541688	361125	180562	90281	45140	22570	7222
	Sound	1512883	239349	957398	718048	478699	239350	119675	59838	29919	9574
	T. Hum.	1615722	218669	874675	656006	437337	218668	109334	54667	27334	8747
	Water	2295288	355579	1422317	1066738	711159	355580	177790	88895	44448	14223
	Subtotal	**11050411**	**1700825**	**6803304**	**5102478**	**3401652**	**1700826**	**850413**	**425206**	**212603**	**68963**
Grand Total		**20780120**	**3073509**	**12294043**	**9220774**	**6147506**	**3074302**	**1539131**	**774009**	**393219**	**137719**
Attack (%)		**46.8%**	**44.7%**	**44.7%**	**44.7%**	**44.7%**	**44.7%**	**44.7%**	**45.1%**	**45.9%**	**49.9%**
Normal (%)		**53.2%**	**55.3%**	**55.3%**	**55.3%**	**55.3%**	**55.3%**	**55.3%**	**54.9%**	**54.1%**	**50.1%**

20% of the processed dataset. The amount of data for each training subset follows these approximated percentages: 0.9%, 2.6%, 5%, 10%, 20%, 40%, 60%, 80%. These percentages have been arbitrarily chosen, and all instances of a small amount are contained in those larger amounts (e.g.: 2.6% has all data of 0.9% and successively). We preserved the class/sensor relation in the subsets except when the number of instances was less or equal to 2000 (threshold). Therefore, the subsets have unbalanced classes as the original dataset, and some classes, such as MITM and Fing have the same records numbers for all subsets.

Table II shows the dataset composition, and the same testing subset was used for all experiments and model evaluation. Additionally, for all models, the validation setting during model fit is 20%, and no normalization (data scaling between 0 and 1) was done. The intention was to test the model's robustness by decreasing the steps and computational work required for the classification. Accuracy, Precision, Recall, F1-Score, and Confusion Matrix are the metrics used during the tests. Those metrics are well-known used and the formulas can be found in related works [7] [17].

We compare the proposed model with two other models. The authors of the Edge-IIoTset dataset claim the first one as the most popular Deep Neural Network (DNN) for cyber-attack detection [5]. The second one is the model from [7] used as inspiration for our proposed model. Due to missing information or details about these two models in their references, we used the best of our judgment to implement them. We believe it is a fair implementation and does not prejudice the model author's work. All the code is publicly available, so other researchers can reproduce our experiments[1].

The output layer's node size and activation function in all evaluated models have been adapted to the evaluation: the output node size is the length of the classes (15 for multiclass or 1 for binary), the `softmax` function was used for multiclass classification, and the `sigmoid` function was used in the case of binary classification.

[1]Repository link: https://phc.st/hybridids

TABLE III: Summary of the results for the eight subsets

Classification Type	Model	Accuracy		Precision		Recall		F1-Score	
		AVG	STD	AVG	STD	AVG	STD	AVG	STD
Binary	Dataset Authors [5]	73.81%	13.66%	83.44%	7.70%	72.07%	12.65%	68.95%	18.79%
	Inspired [7]	**97.85%**	0.18%	**98.13%**	0.15%	**97.59%**	0.20%	**97.81%**	0.19%
	Proposed	**97.85%**	**0.12%**	**98.13%**	**0.10%**	**97.59%**	**0.13%**	**97.81%**	**0.12%**
Multiclass	Dataset Authors [5]	65.96%	11.36%	13.41%	9.62%	15.49%	8.78%	13.45%	7.78%
	Inspired [7]	97.01%	0.76%	81.33%	**7.47%**	70.76%	4.46%	72.82%	**6.42%**
	Proposed	**97.14%**	**0.65%**	**82.32%**	8.06%	**72.66%**	**4.43%**	**74.62%**	7.08%

AVG and STD stand for average and standard deviation, while bolded values highlight the best results.

Two rounds of experiments have been done for each model (proposed, inspired, and Dataset-Authors) per training subset. One for the 15 classes classification and the other for the binary classification performed to all eight training subsets. Consequently, there are 48 (24 for multiclass and 24 for binary classification) results for discussion.

V. Results and Discussion

Table III shows the average results obtained from training and testing in all subsets. Fig. 2 presents the confusion matrix of our proposed model for a selected subset. As the proposed model is based on an existing one, the results of both models are very similar, mainly in binary classification. However, our proposed model showed the best overall results in the multiclass classification. It is important to highlight that our proposed model surpasses the model from the dataset authors in all scenarios.

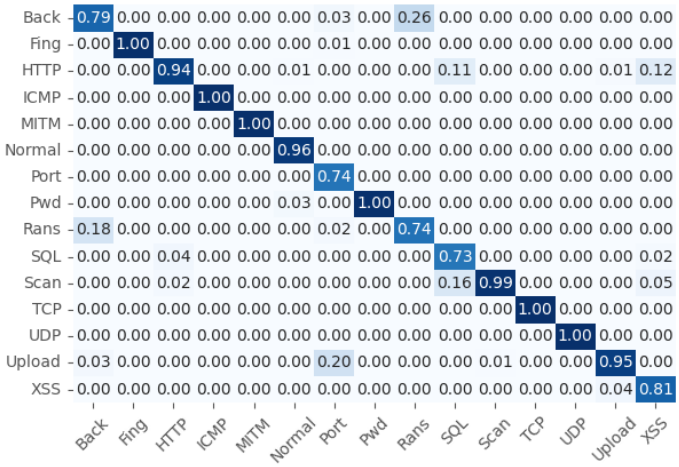

Fig. 2: Confusion Matrix of the proposed model showing the accuracy for 15 classes classification using training subset with 80% of processed data. Axis Y is the real class, and X is the predicted.

Figures 3 and 4 show the evolution of the classification test for each training subset. There is almost no gain in scores with more data training from subset three for binary classification. While the multiclass classification shows a more stable score for accuracy, the other metrics' scores appear to bounce (in both graphics, the macro average value was used for the classes aggregation instead of the weighted average).

The worse model evaluated was the DNN of the author's dataset, which has the max accuracy of 84.7% for the binary classification using 5% and 20% subsets. As this model produced a variety in the results over the eight training subsets,

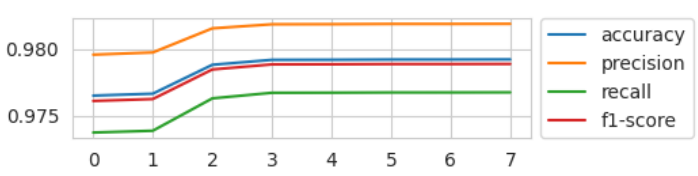

Fig. 3: Proposed model metrics subsets evolution for binary classification. Value 0 to 7 stands for each subset.

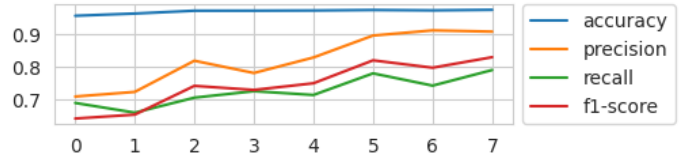

Fig. 4: Proposed model metrics subsets evolution for multi-class classification. Value 0 to 7 stands for each subset.

Fig. 5 shows the results as a boxplot graph. It denotes the poor precision, recall, and f1-score for the multiclass classification.

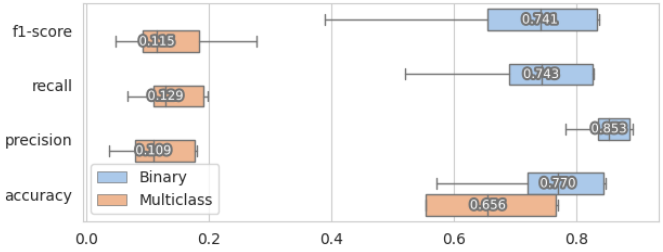

Fig. 5: Dataset authors model distribution and dispersion of eight training subsets results.

It is important to point out that in the results of [5], some classes have low metric score values. Fing, Pwd, Port, SQL, Upload, and XSS attacks have top precision of 71%, due to high relation to the application layer. In our study, those classes were not affected by the exclusion of the application layer, getting the minimal 73% precision for the SQL class between all classes. Besides, the dataset authors mention that to perform the ransomware attack, they explore a backdoor, justifying the confusion between ransomware and backdoor attacks. This same confusion is possible by Back and other HTTP transaction attacks, since the backdoor script is transferred by the `curl` tool generating HTTP traffic. Given that, in our opinion, any model is correct when classifying the traffic in those mixed classes instead of the labelled one (not forgetting the unbalanced dataset situation). This mixed classification between classes related to the HTTP protocol may be hardly seen in our result report but was severally observed over this study, as it can also be seen in the confusion matrix reported by [5]. Moreover, the metrics from [5] must be carefully analyzed since the accuracy and precision metrics mix Attack and Normal classes. Despite this detail in the metrics, in general, the scores from [5] are low compared with our work.

For binary classification (Normal and Attack classes), the proposed model average result for the eight training subsets reached 97.85% in accuracy, while the authors of [5] got 99.99% in the original publication. According to [5], the normal traffic pattern is related to the intrinsic characteristics of IoT devices, allowing a clear distinction and conducive to real-time classification. Nevertheless, supported by our study, the normal traffic of the IoT devices under the application layer can not be 100% distinguished. In our experiments, at least 2% may not be correctly classified. If the input of the proposed model is the Selected ML Dataset file for ML, as proposed by [5], the accuracy for the binary classification is 100%, which leads the thoughts to the next point of discussion.

As reported in [5] and [17], which used the same input, their proposed models got 100% of accuracy, and certainly, it was not caused by the pattern of the normal traffic. It was caused by the application layer features that favor the direct classification. For example, the `mqtt.topic` feature has 'Temperature and Humidity' as value. As another example, it is possible to find an IP address as a value in the `http.referer` feature. Those cases show clearly that the model would learn easily what traffic type is for each recorded sample, automatically leading to the perfect distinction between attack and normal class.

Thus, we argue that there are many advantages to using low layers features instead of application-layer features in traffic automatic classification, as discussed in the next subsections.

A. *Privacy or compliance issues*

The application layer carries more personal data, bringing privacy or law constraints to the discussion of traffic analysis. Cryptography can be applied, but the limitation of some IoT and, more specifically IIoT, devices can restrict their usage. Using non-application layer features makes the easiest path to meet the compliance requirements allowing the users, lawyers, administrators, and all stakeholders to focus on what they need.

B. *Data usage optimization*

The feature selection strategy affects the Machine Learning algorithms' performance. With fewer features, fewer data are used for the model's training and prediction [19]. Consequently, fewer resources like processor clock, memory, cache, bandwidth, and IO operations are required. Making this study proposal attractive for restricted applications or to assemble the solution into devices.

C. *Data generalization*

Using features from the essential packet parts allows a traffic classification model to be more adaptive for different devices, packet extractors, and published datasets, due to the possibility of matching features. In other words, not only IoT/IIoT datasets or data can be used to secure the IoT/IIoT field and vice versa.

D. Data adequacy

Anticipating cases in model development and dataset preparation to emulate a real situation might be challenging when analyzing the available features. Therefore, fewer features mean more predictable values and situations to deal with and be applied to the model. For example, a specific dataset can have some categorical features with values that are not in the real situation or another dataset. Thus, the one-hot processes can change the expected input shape from a model, which will disrupt the prediction. Additionally, when the model can not process the input, normally, a preprocessing step drops this data, suggesting that part of the traffic is not classified in a real-world or real-time application.

E. Model and Application Generalization

Supported by the previously mentioned advantages, a non-application layer trained model would be more suitable for the different network scenarios. It means that based on our approach, the same model can be more robust, being trained by many datasets or real cases and shared between fields of the industry, regardless of whether the data is IoT-based or not. However, it does not prevent having another complementary model to investigate specific attack or network characteristics to attend to the user or business needs.

VI. Conclusion

The main goal of this paper was to answer the question **Is it possible to classify, with high accuracy, the benign and malicious network traffic without specific features from the application layer in an IIoT environment?**, and the results support that **Yes, it is possible**. A large part of a dataset with IoT and IIoT traffic has been analyzed with three neural network models, including one proposed by the authors of the dataset, one previously published in the literature, and a new one proposed by us and inspired by this last one to evaluate this goal. Our model achieved the best results, with 97.85% average accuracy in binary classification and 97.14% average accuracy in multiclass classification. The result discussion highlights five advantages of this approach and the concerns in using the application layer in traffic classification because of privacy and interference by driving prediction. The realistic dataset with distinguished application features is poorly covered in the literature, which makes it difficult to analyze the influence of features from the application layer lacking new research. Also this study advanced the previous studies of Roopak et al. [7] and Ferrag et al. [5] with new analyses and insight. Finally, we published the proposed code for future works and a better understanding of our approach.

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. 18/23098-0.

References

[1] A. Tewari and B. Gupta, "Security, Privacy and Trust of Different Layers in Internet-of-Things (IoTs) framework," *Future Generation Computer Systems*, vol. 108, pp. 909–920, 2020.

[2] Z. Chen, J. Liu, Y. Shen, M. Simsek, B. Kantarci, H. T. Mouftah, and P. Djukic, "Machine Learning-Enabled IoT Security: Open Issues and Challenges Under Advanced Persistent Threats," *ACM Computing Surveys*, p. 3530812, Apr. 2022.

[3] F. Mosaiyebzadeh, L. G. Araujo Rodriguez, D. Macêdo Batista, and R. Hirata, "A Network Intrusion Detection System using Deep Learning against MQTT Attacks in IoT," in *2021 IEEE Latin-American Conference on Communications (LATINCOM)*, 2021, pp. 1–6.

[4] J. Arshad, M. A. Azad, R. Amad, K. Salah, M. Alazab, and R. Iqbal, "A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT," *Electronics*, vol. 9, no. 4, p. 629, Apr. 2020.

[5] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," *IEEE Access*, vol. 10, pp. 40 281–40 306, 2022.

[6] Ferrag, Mohamed Amine, Friha, Othmane, Hamouda, Djallel, Maglaras, Leandros, and Janicke, Helge, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications: Centralized and Federated Learning," Jan. 2022, type: dataset. [Online]. Available: https://ieee-dataport.org/documents/edge-iiotset-new-comprehensive-realistic-cyber-security-dataset-iot-and-iiot-applications

[7] M. Roopak, G. Yun Tian, and J. Chambers, "Deep Learning Models for Cyber Security in IoT Networks," in *Proc. of the IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0452–0457.

[8] J. Alsamiri and K. Alsubhi, "Internet of Things Cyber Attacks Detection using Machine Learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, 2019.

[9] N. Al-Taleb and N. Saqib, "Towards a Hybrid Machine Learning Model for Intelligent Cyber Threat Identification in Smart City Environments," *Applied Sciences*, vol. 12, no. 4, p. 1863, Feb. 2022.

[10] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:," in *Proc. of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116.

[11] H. Hindy, C. Tachtatzis, R. Atkinson, E. Bayne, and X. Bellekens, "MQTT Internet of Things Intrusion Detection Dataset," Jun. 2020, type: dataset. [Online]. Available: https://ieee-dataport.org/open-access/mqtt-internet-things-intrusion-detection-dataset

[12] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, "MQTTset, a New Dataset for Machine Learning Techniques on MQTT," *Sensors*, vol. 20, no. 22, p. 6578, Nov. 2020.

[13] M. Zolanvari, "WUSTL-IIOT-2021," Oct. 2021, type: dataset. [Online]. Available: https://ieee-dataport.org/documents/wustl-iiot-2021

[14] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-IIoTID: A Connectivity- and Device-agnostic Intrusion Dataset for Industrial Internet of Things," Jul. 2021, type: dataset. [Online]. Available: https://ieee-dataport.org/documents/x-iiotid-connectivity-and-device-agnostic-intrusion-dataset-industrial-internet-things

[15] I. Ullah and Q. H. Mahmoud, "An Anomaly Detection Model for IoT Networks based on Flow and Flag Features using a Feed-Forward Neural Network," in *Proc. of the IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 363–368.

[16] "Applications | Research | Canadian Institute for Cybersecurity | UNB." [Online]. Available: https://www.unb.ca/cic/research/applications.html

[17] A. Ghourabi, "A Security Model Based on LightGBM and Transformer to Protect Healthcare Systems From Cyberattacks," *IEEE Access*, vol. 10, pp. 48 890–48 903, 2022.

[18] G. M. Rosa, L. Bonifacio, V. Jeronymo, H. Abonizio, R. Lotufo, and R. Nogueira, "Billions of parameters are worth more than in-domain training data: A case study in the legal case entailment task," 2022. [Online]. Available: https://arxiv.org/abs/2205.15172

[19] O. Aouedi, K. Piamrat, and B. Parrein, "Performance evaluation of feature selection and tree-based algorithms for traffic classification," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.