Evaluation of AI-based Malware Detection in IoT Network Traffic

Nuno Prazeres[1][a], Rogério Luís de C. Costa[2][b], Leonel Santos[1,2][c] and Carlos Rabadão[1,2][d]

[1]*School of Technology and Management (ESTG), Polytechnic of Leiria, Leiria, 2411-901, Portugal*

[2]*Computer Science and Communication Research Centre (CIIC), Polytechnic of Leiria, Leiria, 2411-901, Portugal*

Keywords: Internet of Things, Machine Learning, Intrusion Detection Systems, Cybersecurity.

Abstract: Internet of Things (IoT) devices have become day-to-day technologies. They collect and share a large amount of data, including private data, and are an attractive target of potential attackers. On the other hand, machine learning has been used in several contexts to analyze and classify large volumes of data. Hence, using machine learning to classify network traffic data and identify anomalous traffic and potential attacks promises. In this work, we use deep and traditional machine learning to identify anomalous traffic in the IoT-23 dataset, which contains network traffic from real-world equipment. We apply feature selection and encoding techniques and expand the types of networks evaluated to improve existing results from the literature. We compare the performance of algorithms in binary classification, which separates normal from anomalous traffic, and in multiclass classification, which aims to identify the type of attack.

1 INTRODUCTION

The Internet of Things (IoT) is one of the drivers for a new generation of communication networks, combining a wide variety of hardware and software that provide easy-to-use and low-cost solutions to customers. IoT devices perform several critical tasks, and collect and share large volumes of data, including private ones. But without security measures, such devices may be vulnerable to attacks that compromise users' privacy and the behavior and availability of services.

In the last few years, machine learning methods have been used to analyze large data volumes in diverse environments, correlating events, identifying patterns, and detecting anomalous behavior (Berman et al., 2019) that, otherwise, would remain hidden. In this work, we deal with using machine learning methods to identify malign traffic in the IoT. We apply machine learning on statistics about traffic flow which is a more scalable and interoperable approach than analyzing packets payload (Santos et al., 2021). Hence, traffic data must be aggregated into flow information before looking for anomalous traffic.

We evaluate the use of Supervised Learning methods, including deep networks. A key challenge when using Supervised Learning is to get large labeled datasets representing the analyzed phenomenon. Although there are several works on the use of machine learning for intrusion detection in IoT, most of the current evaluations does not use data on real IoT traffic, instead they use more general network traffic datasets like UNSW-NB15, NSL-KDD and KDD99 (Ashraf et al., 2021; Moustafa and Slay, 2015).

In this work, we evaluate the learning approaches on data of the IoT23 dataset (Garcia et al., 2020), which contains benign and malign traffic from real-world IoT equipment. Austin (Austin, 2021) used the IoT-23 dataset for binary classification, i.e., to separate anomalous traffic from the normal one. We expand the number of evaluated algorithms (including deep networks) and use distinct encoding and feature selection methods than previous work to achieve higher identification performance. We also evaluate multiclass classification that aim to identify the type (class) of attack. The main contributions of this work include an evaluation using traffic data from real IoT equipment, and the assessment of traditional machine learning methods and deep networks for binary and multiclass classification in such context.

The following section reviews background and related work. In Section 3 we describe the dataset and the pre-preprocessing and feature selection strategies. Section 4 contains experimental results. Section 5 presents conclusions and future works.

[a] https://orcid.org/0000-0003-1760-6220
[b] https://orcid.org/0000-0003-2306-7585
[c] https://orcid.org/0000-0002-6883-7996
[d] https://orcid.org/0000-0001-7332-4397

Prazeres, N., Costa, R., Santos, L. and Rabadão, C.
Evaluation of AI-based Malware Detection in IoT Network Traffic.
DOI: 10.5220/0011279600003283
In *Proceedings of the 19th International Conference on Security and Cryptography (SECRYPT 2022)*, pages 580-585
ISBN: 978-989-758-590-6; ISSN: 2184-7711

2 BACKGROUND

Cybersecurity systems can be used together with machine learning to take advantage of its characteristics to develop increasingly robust attack detection methods and solutions for IoT.

2.1 Intrusion Detection Systems

An Intrusion Detection System (IDS) is a tool or mechanism used to detect non-authorized accesses and attacks against systems, analyzing the communications, internal activities, and other events. The IDS aims to detect any anomaly or attack in real-time and uses the network traffic as its data source. These systems work with different detection methods and must be strategically placed in the network.

A flow-based IDS analyzes traffic information and statistics rather than packets payload. Sensors are responsible for capturing data from packets transmitted in the reference, which must then be aggregated and transformed into data streams. Such flows have a series of statistics and characteristics, as defined in the Internet Protocol Flow Information Export (IPFIX) (Claise, 2008; Claise et al., 2008; Claise and Trammell, 2013). Using packet flows in IDS for IoT can make these solutions more scalable and interoperable (Santos et al., 2021). In this paper, we use flow data to classify traffic and identify anomalous traffic and attacks. The dataset we evaluate contains network flow data resulting from several anomalous traffic and malware, including DDoS attacks, C&C activity and bot activity to hijack insecure devices.

2.2 Machine Learning

Machine Learning techniques have been used in several contexts where there is a need to process and analyze large volumes of data, spotting patterns, behaviors, and anomalies inside the datasets. In this work, our models classify network traffic flows. In classification tasks, the model tries to identify rules from the sample data and predicts the belonging of new elements (objects, individuals, and criteria) to a given class (Hussain et al., 2020).

In Supervised Learning, the model learns from labeled data, which means that training data includes both the input and the desired results (Chaabouni et al., 2019). A key challenge when using Supervised Learning is to get large labeled datasets representing the analyzed phenomenon.

Algorithms like Logistic Regression (LR) and Naïve Bayes (NB) belong to the family of probabilistic classifiers. In the Decision Trees (DT) algorithm, a tree has several node leaves or decision nodes that classify the data. The Random Forest (RF) aggregates several trees that work independently with a similar data entry, assembling a committee of trees. This correlation of classifiers will make a more resilient model, protecting the trees of individual errors.

Artificial neural networks (ANNs) are generic algorithms mimicking the biological functioning of a brain without being intended for a specific task (Chaabouni et al., 2019). A Multilayer Perceptron (MLP) is one of the simpler ANN. The MLP has an input layer that receives data, an output layer that outputs the decision or prediction about the input, and an arbitrary number of hidden layers that are formed by interconnected neurons and are the computational engine of the MLP. In Deep Learning (DL), ANN learns to represent the data as a nested hierarchy of concepts within the layers of the neural network (Chalapathy and Chawla, 2019), which may leads to superior performance over traditional machine learning in large datasets (Al-Garadi et al., 2020)

2.3 Related Work

Zeadally & Tsikerdekis (Zeadally and Tsikerdekis, 2020) discuss the use of traditional network monitoring (like Intrusion Detection Systems) with the help of machine learning algorithms to give a viable alternative to existing IoT security solutions. They summarize the needs of host and network-based approaches to perform network traffic capture using machine learning to process the data. In this case, no option is given as optimal, highlighting only the strength and limitations of the machine learning algorithms regarding the IoT devices characteristics.

Chaabouni et al. (Chaabouni et al., 2019) present a comprehensive survey pointing out the design challenges of IoT security and classification of IoT threats. As future research directions, the authors state that exploring the edge and fog computing paradigms would give the ability to push the intelligence and processing logic employment down near to data sources. Authors identify that current works use datasets like UNSW-NB15 (Moustafa and Slay, 2015), NSL-KDD (Moustafa and Slay, 2015), and KDD99 (Cup, 1999), but there is a lack of works that use real-world IoT-dedicated dataset to train and deploy an IoT IDS based on machine learning.

Austin (Austin, 2021) used the IoT-23 dataset to evaluate binary classification. Nevertheless, the author uses the timestamp of network traffic as a top feature to train the networks. Here we use a distinct set of features, and we also evaluate the use of deep models and multiclass classification.

3 THE IoT-23 DATASET AND FEATURE SELECTION

The IoT-23 dataset (Garcia et al., 2020) results from the Malware Capture Facility Project from the Czech Technical University ATG Group and contains normal and malicious traffic. Real hardware (not simulated), including a smart door lock (Somfy), a smart LED lamp (Philips), and a home intelligent personal assistant (from Amazon), is used to produce benign traffic. Network attacks are mainly based on known botnets like Mirai and in trojan software that helps the malicious actors take over the equipment remotely.

The flows were captured through a passive open-source network traffic analyzer called Zeek, formerly known as Bro. Zeek's main data structure is a connection that follows typical flow identification mechanisms, such as 5-tuple approaches, this structure consists of the source IP address/port number, destination IP address/port number, and the protocol in use.

The provided dataset contains 21 columns representing the communications made inside the network and two columns that label each flow as benign or malicious and identify the attack type of malicious flows.

Originally, traffic flows were split into log files accordingly to the malware type. To build a sample dataset representing the various types of network attacks produced in the lab, we took samples from several log files. Our final dataset has 1,244,220 flows, with the total number of malicious flows near the number of normal flows. Table 1 presents the number of flows in the dataset per class.

Table 1: Multiclass label encoder.

Class	Description	#Flows
0	-	622.110
6	PartOfAHorizontalPortScan	522.896
5	Okiru	86.360
3	DDoS	10.277
2	C&C	2.428
1	Attack	140
4	FileDownload	9

We pre-processed the data on the sample dataset, transforming the features from categorical (strings) to numerical values and executing a normalization. We used the LabelEncoder (Pedregosa et al., 2011) and the SimpleImputer (Pedregosa et al., 2011), to transform strings into values, and the StandardScaler (Pedregosa et al., 2011) algorithm for normalization.

Feature selection is a key activity in machine learning with a great impact on the performance of the ML models. In this process, features are (automatically or manually) selected based on their contribution to predicting a variable or output in which we are interested. Including irrelevant features in the input data can impact negatively on performance.

With feature selection in mind, we initially removed from the sample dataset, the columns with a single value (i.e., *local_orig* and *local_resp*) and columns with unique values at each row (i.e., *ts* and *uid*). We assume that those columns have no significance in determining if a flow is malicious. We also removed the *tunnel-parents* column as it is related to the *uid* column, which was dismissed. Then, the sample dataset remains with 16 columns describing flow characteristics and the 2 columns that label the traffic.

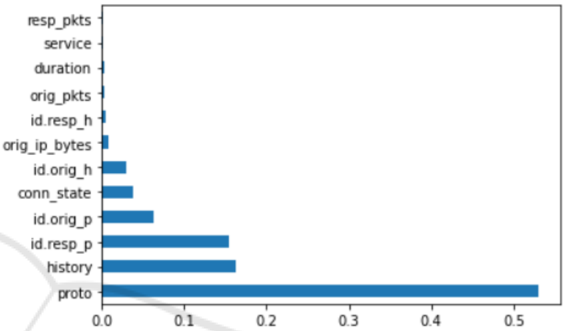


Figure 1: ExtraTree Classifier feature selection.

To verify which of the remaining features have bigger contribution for flow classification we used the ExtraTreeClassifier (Pedregosa et al., 2011). Figure 1 presents the computed influence of each feature. We selected the features that contributed with a score grater than 0.05 (namely, *proto*, *history*, *id.resp_p* and *id.orig_p*) to keep in the sample dataset that would be used to train and test machine learning algorithms.

4 EXPERIMENTAL EVALUATION

We validated our proposals executing binary and multiclass classification on real-world IoT network flows. We used Python (Van Rossum and Drake, 2009), Jupyter Notebooks (Kluyver et al., 2016), and the IoT23 dataset (Garcia et al., 2020) to train and several machine learning algorithms, including Logistic Regression, Random Forest, Naïve Bayes and a Multiplayer Perceptron.

We used scikit-learn (Pedregosa et al., 2011) to run the Logistic Regression, Random Forest, and Naïve Bayes algorithms. We used Tensorflow and Keras to build a dense Multilayer Perceptron with four hidden layers and fully connected neurons.

4.1 Performance Metrics

Using adequate metrics is of major importance when assessing model performance. The most suitable indicator depends on the problem of interest. For instance, in this work, we use machine learning to classify network flows into normal or malicious. A binary classification problem has four possible outcomes. The correctly predicted negatives are the *true negatives* (TN), and the correctly predicted positives are *true positives* (TP). The incorrectly predicted negatives are the *false negatives* (FN). Finally, the incorrectly predicted positives are called *false positives* (FP). We use confusion matrices to summarize the values of TN, TP, FN and FP.

One of the commonly used metrics is *precision*, which measures how accurate the classification model is. Precision is defined on the number of correctly classified elements, as represented in Equation 1.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

In practice, misclassifications may have different importance. For instance, classifying a malicious flow as a normal one may be more prejudicial than identifying a normal flow as a malicious flow. The TP rate - i.e., *recall* - depends on the number of true positives and false negatives is defined by Equation 2.

$$Recall = TPR = \frac{TP}{TP + FN} \tag{2}$$

Models with high precision and recall values are highly dependable, as they do not misclassify benign flows and do not wrongly leave out malicious flows. On the other hand, models that achieve high precision values but low recalls miss out on many malicious flows. Therefore, these models should not perform critical tasks. Lastly, models with high recall and small precision values would detect most of the malicious flows but also raise many false alarms, which can create entropy in the security system.

The *F1 Score* (or *F-score*) is useful to evaluate the performance of models on unbalanced datasets. It is the harmonic mean of the precision and recall, as defined in Equation 3.

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

4.2 Classification Results

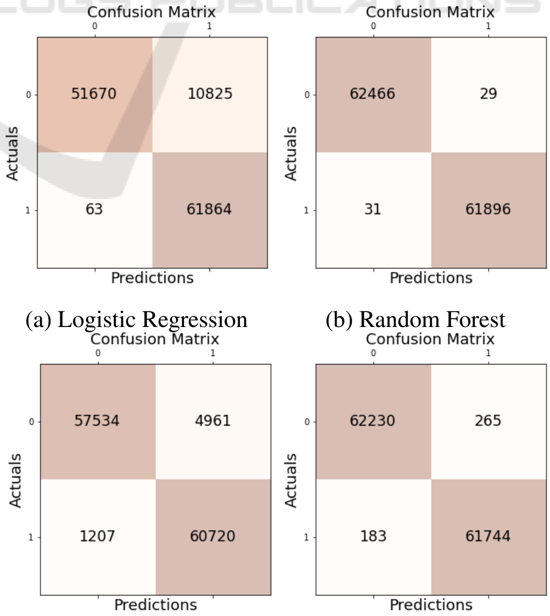
We evaluated the models both for binary classification and multiclass classification. In binary classification, the model classifies each network flow as a normal or anomalous one. In the multiclass classification, the training data contains an identifier for each attack and, besides identifying which flows correspond to attacks, the classifier should also identify the type of attack for each flow it classifies as anomalous. Our deep network has four hidden layers and fully connected neurons. We used 87.5% of the flows for model training and the remaining for testing trained models. To avoid overfitting, we split training data into train and validation sets and used Early Stopping.

4.2.1 Binary Classification

Table 2 the values of precision and recall for binary classification. Logistic Regression (LR) achieved the lowest precision values, but it still got a high recall. The deep network (ANN) and the Random Forest (RF) algorithm got high values for precision and recall. Our results outperformed the ones of the literature in all metrics and methods, except for the recall of Naïve Bayes (NB).

Table 2: Binary classification comparison.

Models	Our results		Results from (Austin, 2021)	
	Recall	Precision	Recall	Precision
LR	0.9990	0.8511	-	-
RF	0.9995	0.9995	0.9498	0.9992
NB	0.9805	0.9245	0.9975	0.8976
ANN	0.9970	0.9957	-	-



(a) Logistic Regression (b) Random Forest

(c) Naïve Bayes (d) Multi-layer ANN

Figure 2: Confusion Matrices - Binary Classification.

Figure 2 presents the Confusion Matrix for each evaluated model (0 = normal traffic, and 1 = malicious traffic). Such matrices show that Naïve Bayes was the method with the worst results in terms of classifying anomalous traffic as a normal one.

4.2.2 Multiclass Classification

In Multiclass Classification, we evaluated the models' performance on classifying network flows as a normal flow or as belonging to one of the seven classes of attack in the IoT23 dataset. The testing with the traditional models reveals the struggle of accurately classifying the flows. There are no flows of class 4 in the test set, still, the Random Forest classifier identified one normal flow as being of class 4.

Although the dataset is balanced in terms of benign and malign traffic, it is unbalanced in the types of malicious flows. Hence, in terms of performance metrics, we also computed the F1-Score for each evaluated method. The Random Forest (Table 5) algorithm and the Multi-layer ANN (Table 6) achieved the highest F1-score values. Logistic Regression (Table 3) and Naïve Bayes classifiers (Table 4) got low precision and recalls values for classes 1 and 2. Indeed, the Logistic Regression algorithm totally failed to identify flows of classes 1 and 2 in the testing set. Naïve Bayes also got low performance for attacks of class 3.

Table 3: Logistic Regression - Multiclass Classification.

Class	Precision	Recall	F1-Score
0	0.9372	0.8618	0.8978
3	0.8924	0.9849	0.9363
5	0.9796	0.8790	0.9266
6	0.8523	0.9514	0.8991

Table 4: Naïve Bayes - Multiclass Classification

Class	Precision	Recall	F1-Score
0	0.9976	0.9327	0.9641
1	0.2500	1.0000	0.4000
2	0.0290	0.2760	0.0525
3	0.5503	0.8863	0.6790
5	1.0000	0.9986	0.9993
6	0.9603	0.9792	0.9697

Table 5: Random Forest - Multiclass Classification.

Class	Precision	Recall	F1-Score
0	0.9995	0.9994	0.9994
1	1.0000	0.9333	0.9655
2	0.9908	0.9908	0.9908
3	1.0000	1.0000	1.0000
5	1.0000	1.0000	1.0000
6	0.9993	0.9994	0.9993

Table 6: Multi-layer ANN - Multiclass Classification.

Class	Precision	Recall	F1-Score
0	0.9969	0.9978	0.9973
1	0.6250	0.3333	0.4348
2	0.8000	0.5530	0.6540
3	1.0000	1.0000	1.0000
5	0.9991	1.0000	0.9995
6	0.9936	0.9963	0.9963

4.3 Discussion

The results obtained for the binary classification evaluation confirm the feasibility of most of the evaluated models. The Random Forest algorithm and the multi-layer neural network got good performance, and outperformed previous work.

Multiclass classification metrics were significantly worst than binary classification. Most of the methods got their worst results when classifying flows of classes 1 and 2. As the number of training samples for flows of these classes is much lower than for other ones (Table 1), the poor performance of the various methods when analyzing data from these classes demonstrates the importance of the test sample in choosing supervised learning methods. Still, it is worth noting that the training base contains more than 2.000 flows of class 2, which leads to the need for a high number of training samples to achieve high identification performance.

Obtained results and the need for a high number of sample flows of each class for training the models indicate that binary classification may be the most viable strategy to identify anomalous traffic. Still, a possible future work would be to evaluate the application of multiclass identification only on flows that binary classification methods identify as malign.

5 CONCLUSIONS

The large-scale use of IoT devices enabled the implementation of several smart services and environments.In this context, devices collect and share a large amount of data, including private data. It is an attractive environment for potential attackers, and it must have its cyber protection measures, such as a specialized flow-based IDS. On the other hand, machine learning techniques have recently been adopted in several areas to handle a large volume of data, learn patterns, and use the acquired knowledge to classify new elements. Still, there are just a few datasets with benign and malign traffic of real-world IoT networks. In this work, we evaluated machine learning to classify anomalous traffic in real-world IoT traffic flows.

We assess several methods, from traditional techniques with supervised learning to deep neural networks. We initially performed the binary classification of traffic flows, where the system classifies each new flow into normal or anomalous. The random forest algorithm and the multilayer neural network achieved the best (and satisfying) performance values.

We also evaluated a multiclass classification approach, on which the classifier should identify the type of attack of each flow it classifies as anomalous. The results in this approach were considerably worse than the ones we got with binary classification. Although the training and test sets are balanced in terms of benign and malign traffic, they were unbalanced in the types of malicious flows and some methods failed when identifying some types of malign traffic. Even though the training set counts with thousands of samples of one of such traffic, the relatively small number of samples available for training had negatively impacted the performance of the models. Still, a possible future work would be to evaluate the application of multiclass identification methods only on flows that binary classification methods identify as malign.

As future work, we also intend to expand the analysis of deep models with greater capacity to identify temporal patterns and evaluate model resilience to adversarial machine learning.

ACKNOWLEDGEMENTS

This work is funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., under the Scientific Employment Stimulus - Institutional Call - CEECINST/00051/2018 and in the context of the project UIDB/04524/2020.

REFERENCES

Al-Garadi, M. A., Mohamed, A., Al-Ali, A. K., Du, X., Ali, I., and Guizani, M. (2020). A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys and Tutorials*, 22(3):1646–1685.

Ashraf, J., Keshk, M., Moustafa, N., Abdel-Basset, M., Khurshid, H., Bakhshi, A. D., and Mostafa, R. R. (2021). Iotbot-ids: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities. *Sustainable Cities and Society*, 72:103041.

Austin, M. (2021). IoT Malicious Traffic Classification Using Machine Learning. Master's thesis, Statler College of Engineering and Mineral Resources - West Virginia University.

Berman, D. S., Buczak, A. L., Chavis, J. S., and Corbett, C. L. (2019). A survey of deep learning methods for cyber security. *Information (Switzerland)*, 10(4).

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., and Faruki, P. (2019). Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Communications Surveys and Tutorials*, 21(3):2671–2701.

Chalapathy, R. and Chawla, S. (2019). Deep Learning for Anomaly Detection: A Survey. *arXiv preprint*, pages 1–50.

Claise, B. (2008). Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 5101.

Claise, B., Quittek, J., Meyer, J., Bryant, S., and Aitken, P. (2008). Information Model for IP Flow Information Export. RFC 5102.

Claise, B. and Trammell, B. (2013). Information Model for IP Flow Information Export (IPFIX). RFC 7012.

Cup, K. (1999). Data/the uci kdd archive, information and computer science. *University of California, Irvine*.

Garcia, S., Parmisano, A., and Erquiaga, M. J. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic.

Hussain, F., Hussain, R., Hassan, S. A., and Hossain, E. (2020). Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Communications Surveys and Tutorials*, 22(3):1686–1721.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. and Schmidt, B., editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press.

Moustafa, N. and Slay, J. (2015). Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Santos, L., Gonçalves, R., Rabadão, C., and Martins, J. (2021). A flow-based intrusion detection framework for Internet of Things networks. *Cluster Computing*, pages 1–21.

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Zeadally, S. and Tsikerdekis, M. (2020). Securing Internet of Things (IoT) with machine learning. *International Journal of Communication Systems*, 33(1):e4169.