Machine Learning and Visualization tools for Cyberattack Detection

Robin Duraz, David Espes, Julien Francq, Sandrine Vaton

Machine Learning and Visualization tools for Cyberattack Detection

Robin Duraz[*⋆], David Espes[♯], Julien Francq[‡], Sandrine Vaton[⋆]
[*]*Chaire of Naval Cyber Defense, Ecole Navale,* [♯]*Université de Bretagne Occidentale,* [‡]*Naval Group (NCL),*
[⋆]*IMT Atlantique, Lab-STICC (UMR 6285)*
[*]Lanvéoc, [♯⋆]Brest, [‡]Ollioules (France)
surname.name@{[*]ecole-navale.fr, [♯]univ-brest.fr, [‡]naval-group.com, [⋆]imt-atlantique.fr}

***Abstract*—As technology develops and pervades our world, IT threats are becoming more and more common. Cyberattacks, while relatively rare a decade ago, are nowadays occurring much more frequently, putting at risk various institutions, ranging from a simple hospital to big companies. While it is necessary to secure a system, attackers are always finding new ways to circumvent security measures, thus motivating the use of Intrusion Detection Systems (IDS) to detect cyberattacks. In this work, results obtained by using Machine Learning (ML) algorithms to detect cyberattacks in a public dataset, and visualization tools that can provide a subjective assessment of the task difficulty and the ML model quality are presented.**

***Index Terms*—Machine Learning, cybersecurity, classification, visualization, data representation**

I. Introduction

The world as a whole is being more and more digitized. Industries especially, whether it be to reduce costs, increase performance, or even simply respond to various needs and obligations, transition to more automated and connected systems. Industrial control equipments that were rather autonomous, isolated and generally required physical operation are getting more and more connected through IT equipments, allowing for easier monitoring and control. With that come new security threats that targeted IT equipments and can now also reach OT equipments through them.

Obvious ways to make the system more secure is to consider the CIA triad (Confidentiality, Integrity and Availability) of Information Technology, and propose systems that are by design more secure [1]. Nevertheless, no system is ever perfectly secure, so complementary methods are needed to detect attacks when they occur. It is thus important to build IDSs that will be the last line of defense against cyberattacks.

Typical cybersecurity approaches mainly use static methods, based on signatures, to identify potential threats. These approaches work relatively well for known and easy to identify threats, but quickly become limited when faced with evolving malware and other unknown threats. ML methods can be more flexible tools in this changing context. Binary classification presenting some limitations, the main goal of this work[1] is to present the results that can be expected from ML algorithms performing multi-class classification on a public cybersecurity dataset. It also shows the use of visualization tools to estimate the difficulty of the task and the performance of the algorithms used.

Section II of this paper presents works relevant to the use of ML for cybersecurity. Section III discusses the problem and the proposed approach, with results discussed in Section IV, before conclusions.

[1]Funded by the Chair of Naval Cyber Defence[*] and its partners Thales, Naval Group, French Naval Academy, IMT-Atlantique, ENSTA-Bretagne and Region Bretagne.

II. Related work

A. ML applied to cybersecurity

ML applied to cybersecurity, and more specifically to cyberattack detection is a relatively new area of research, and one that is constantly evolving. [2] [3] [4] painted the landscape of what is achievable by applying ML algorithms to cybersecurity problems, with supervised and unsupervised approaches, but lack information about results on more recent datasets and/or multi-class classification.

In a real world environment, normal traffic is predominant and cyberattacks are relatively rare. This is fortunate, but at the same time makes it more difficult for ML models relying on statistics to differentiate them. When considering multiple classes representing different attacks, the task is increasing further in difficulty.

While the task might be easier when wanting to differentiate any attack from normal traffic without differentiating them, the end goal of using the attack class for mitigation purposes can motivate the idea of multi-class classification, in order to have faster and more appropriate responses. It also helps in avoiding some pitfalls, like the fact that performance on a much more frequent attack (DDoS for example) can hide poor performance on other attacks.

B. Cybersecurity datasets

Finding a good cybersecurity dataset can often be a challenging task. Being able to work on public data obtained from a real cyberattack would assuredly be the best solution, but it is often not doable for privacy and security reasons. Another solution is to simulate either the environment, the attacks or even both, while still trying to make it as realistic as possible. Although it is easy to get data that way, those datasets often suffer from multiple limitations, such as a simplistic environment or the lack of diversity in the attacks realized.

Some well-known public datasets include (NSL-)KDD'99, ISCX 2012, CTU-2013, UNSW-NB15, and CICIDS2017 [5]. In this paper, the CICIDS2017 dataset is used, being quite complete in both the simulated environment and the cyberattacks performed. Table I describes the attacks.

TABLE I
CICIDS2017 ATTACKS

Attack name	**Description**
Botnet (Ares)	Remote shell, keylogging and others
DDoS	Junk TCP, UDP and HTTP GET requests
DoS GoldenEye	Uses HTTP KeepAlive and NoCache
DoS Hulk	Dynamic requests
DoS Slowloris DoS Slowhttptest	Keep connection open by continuously sending small packets
FTP/SSH-Patator	Brute force attack over FTP/SSH
Heartbleed	Attack on a vulnerable SSL version
Infiltration	Uses an infected dropbox file or USB key to perform a portscan attack
Portscan	Nmap with various options, sS, sT, sF, etc.
Web Attack Brute Force / SQL Injection / XSS	Performed on a vulnerable PHP/MySQL Web App

The resulting dataset is composed of around 2.8 million samples with 78 features that are metadata about the flow's statistics, such as Flow Duration, Min Packet Length, Mean Packet Length, etc.

III. Proposed approach

To perform multi-class classification, the label used is a word corresponding either to "benign" for normal traffic or the attack name. As a result, the ML model predicts one of the 15 labels for each sample that composes the dataset.

A. Data description and processing

Preprocessing is performed on the CICIDS2017 dataset to remove samples containing infinite or NaN values. Labels are changed from a word to a corresponding integer. The dataset was then split into a training set (70% of the dataset) and testing set (the remaining 30%) using a random split, only ensuring that both sets contain all classes.

B. Classification methods

For the purpose of understanding what kind of performance could be expected from ML algorithms in multi-class classification to detect cyberattacks, multiple algorithms are tested.

In the unsupervised setting, the K-means algorithm is tested. Label information was not used at training time but is used during testing to compare its efficiency with those of other models.

For supervised approaches, Decision Trees (DTs), Random Forests (RFs), Linear SVC (LSVC), Logistic Regression (LR), Gaussian Naïve-Bayes (GNB), Multi Layer Perceptron (MLP) algorithms are used from the `scikit-learn`[2] library, while the Deep Neural Network (DNN) algorithm used the `PyTorch`[3] library.

[2]https://scikit-learn.org/stable/index.html
[3]https://pytorch.org/

All models are tested with various hyperparameters and architectures in order to find the best performance. It is also important to note that data is min-max normalized for all models except for DTs and RFs.

C. Visualization algorithms

The idea of using visualizations came from the fact that high dimensional data is difficult to make sense of. Creating low dimensional visualizations could help to understand the limits of the models and if data is the limiting factor.

IV. Results and Discussion

Here, a comparison of the different models' performance is presented, as well as what can be inferred from visualizations.

A. Metrics

In order to compare the performance of different models, multiple common ML metrics are used, like accuracy (number of correctly classified instances over the number of instances), precision, recall, and F1-score (dealing with True Positives, True Negatives, False Positives and False Negatives). The latter three are computed for all the models except for DNNs. We denote further by $Attacks \leq X\%$ the number of attack classes that have an accuracy lower than X%. The accuracy used is the same as before, considered class by class.

Accuracy, precision, recall and F1-score were computed over the whole dataset as well as for each class.

B. Model performance

A summary of model performance showing accuracy values can be found in Table II. Mean accuracies, precision, recall and F1-score are rather high for all models. It is mainly due to their ability to correctly classify benign traffic. However, the dataset is significantly unbalanced because the benign traffic accounts for 80% of the whole traffic, so a more interesting approach is to consider class-related measures, of which columns 3 to 5 show that it is possible to have more than 90% accuracy while having poor accuracies on many attacks.

In half of the models, the performance is good for benign traffic, but significantly lower for many attack classes. This is actually concerning because it means the models will miss the attacks that are occurring and thus not raise any alert, endangering the system. In the other half, the performance seems much better, although some attacks are still missed. While performances can be improved, it is also possible that the data used is simply lacking fine-grained information for the models to attain higher scores.

Deciding which algorithm has the best results might depend on the operational needs. A lower accuracy on benign traffic means that part of it will be misclassified as attacks and thus raise false alarms, while lower accuracy on attack classes means attacks will be missed. Benign traffic being predominant, raising too many false alarms can be problematic. Though some differences (normalization, for example) can impact the performance of other models, the best algorithm overall seems to be DTs, raising few false alarms and missing

TABLE II
MODEL PERFORMANCES

Model	Accuracies					F1-Score
	Mean	Benign	attacks $\leq$ 90%	attacks $\leq$ 50%	attacks $\leq$ 10%	
GNB	0.8	**1.00**	14	14	14	0.72
LR	0.94	0.98	13	8	8	0.93
LSVC	0.93	0.98	12	6	6	0.93
K-Means	0.95	0.97	13	6	6	0.95
MLP	0.97	0.98	5	5	3	0.97
RF	**0.99**	0.99	5	2	0	**0.99**
DT	**0.99**	0.99	5	**1**	**0**	**0.99**
DNN	0.95	0.94	**4**	2	1	/

*Values were truncated to the second decimal

less attacks than other models. Another consideration to take is that, although more interpretable, DTs are quite prone to overfitting, so it would be safe to verify that it is not the case here.

C. Data visualization

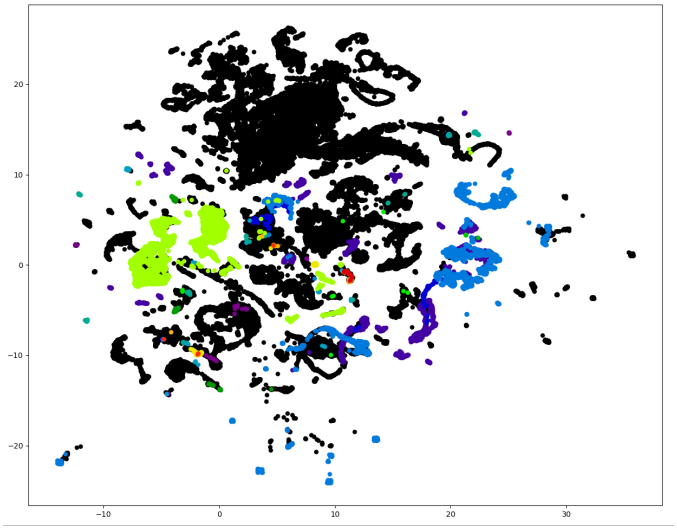

Fig. 1. UMAP visualization on 10% of the test dataset. ● benign traffic, ● Botnet, ● DDoS, ● DoS GoldenEye, ● DoS Hulk, ● DoS slowhttptest, ● DoS slowloris, ● FTP-Patator, ● Heartbleed, ● Infiltration, ● Portscan, ● SSH-Patator, ● Web Attack Brute Force, ● Web Attack SQL Injection, ● Web Attack XSS

Data visualization serves multiple purposes. First, it could help gauging the difficulty of the classification task and give an idea of the expected performance of ML models. The more separated the different classes are in the visualization (that only uses notions of distance), the easier the task would be. Secondly, it can give an idea of how good are different DNN architectures, by showing how separated are the classes in their inner representation. It can also be compared to the original data, to consider if DNNs inner representations might be more suited in case of new classes appearing.

Figure 1 is done with the Uniform Manifold Approximation and Projection (UMAP [6]) algorithm using only part of the test dataset. This algorithm tries to reduce data dimension while retaining topological structure. By performing visualizations with different amounts of data, it could be seen that the more data is used, the more benign traffic is omnipresent and difficult to separate from other classes. It also helped selecting better DNN architectures by comparing visualizations of their last hidden layer.

V. CONCLUSION AND FUTURE WORK

In this paper, a basic overview of the performance of ML models on the CICIDS2017 dataset is presented. Visualization algorithms are also presented as a tool to better understand the complexity of the dataset as well as what can be expected from different NN architectures.

Interestingly enough, the best performing algorithm is DT, which is similar to RF, but simpler. One of the main drawbacks of supervised approaches is that they are unable to handle classes that are not seen during training, thus limiting their use in real situations, where attacks such as zero-day attacks are particularly dangerous. They would need to be regularly trained again to include new classes. Unsupervised approaches could be an answer to this problem, but their performance is still lacking.

As future work, we plan to focus on researching solutions to better handle unknown attack classes. We also plan to explore Explainable Artificial Intelligence (XAI) which can provide better answers to explain a model prediction compared to simply visualizing data.

REFERENCES

[1] Y. Ashibani and Q. H. Mahmoud, "Cyber Physical Systems Security: Analysis, Challenges and Solutions," *Computers and Security*, vol. 68, pp. 81–97, 2017.

[2] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35 365–35 381, 2018.

[3] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity Data Science: an Overview From Machine Learning Perspective," *Journal of Big Data*, vol. 7, no. 1, 2020.

[4] J. Meira, R. Andrade, I. Praça, J. Carneiro, V. Bolon-Canedo, A. Alonso-Betanzos, and G. Marreiros, "Performance Evaluation of Unsupervised Techniques in Cyber-Attack Anomaly Detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4477–4489, 2019.

[5] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018.

[6] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform Manifold Approximation and Projection for Dimension Reduction," *CoRR*, 2018.