Exploring the use of machine learning outlier detection algorithms for DDoS identification

André L. Ribeiro, Othávio R. C. Araújo, Caio A. C. Maciel, Leonardo B. Oliveira

Computer Science Department (DCC)
Federal University of Minas Gerais (UFMG) – Belo Horizonte – MG – Brazil

{andre.ribeiro, othavio.rudda, caio.campos, leob}@dcc.ufmg.br

Abstract. *Servers and users rely on safe defenses against multiple attacks. Usual practices, however, normally are unable to deal with huge distributed attacks, such as DDoS. This is a malicious practice that aims to interrupt the flow of a network causing data congestion. Moreover, DDoS is a stealthy practice, as its traffic might present similar attributes to usual ones. With this in mind, in this paper, we use unsupervised, semi-supervised, and supervised machine learning algorithms to automatically analyze a selected network, detecting possible DDoS flows using PyOD library. We evaluate each of those types of algorithms and also explore the effects of previous feature selection on them.*

1. Introduction

Distributed Denial of Service (DDoS) is a malign practice developed with the maturing of Denial of Service (DoS). Those two attacks have as goals promoting interference in CPU time and consuming network resources, generating a flood of requests in the victim's network, as described by Sonar and Upadhyay [Sonar and Upadhyay 2014]. Furthermore, DDoS is a kind of DoS attack that can originate from several different IP addresses and is therefore called distributed DoS.

The first known DDoS attack, named "trinOO" [1], happened in 1999 when the attacker sent a huge series of UDP packets to the University of Minnesota servers, bringing down their network for two days. These events became more frequent through the years, but the biggest reported attack happened in 2018 against the popular GitHub[2] website that was down for 6 minutes after dealing with 1.35 terabits per second of traffic.

The recognition of DDoS attacks is a type of outlier detection because it aims to detect an atypical behavior in a data flow. To mitigate this kind of problem, it is possible to use manual analysis observing network patterns in an extremely expensive process. Otherwise, this problem can be prevented using automatic machine learning processes as presented in this work.

Specially designed for outlier detection in Python, PyOD library [Zhao et al. 2019] implements and gathers some machine learning methods for this kind of task. The library includes algorithms of five areas: linear models, proximity-based, probabilistic, outlier ensembles, and neural networks. Given the usual nature of outlier detection tasks and labeled data availability, it focuses on unsupervised and

[1] trin00 incident report: http://www.cert.org/incident_notes/IN-99-04.html. Accessed in 15 jul 2020.
[2] GitHub incident report: https://github.blog/2018-03-01-ddos-incident-report/. Accessed in 15 jul 2020.

semi-supervised learning, however, it also implements one algorithm of supervised learning. Most methods presented in this work are implemented by PyOD.

Contributions. The main contribution of this work is the evaluation and comparison of some unsupervised, semi-supervised, and supervised machine learning algorithms for outlier detection applied to DDoS identification. The specific contributions are:

1. The identification of DDoS attacks in the given dataset through different methods.
2. A comparison of the efficiency of some PyOD implemented methods in terms of computational resources required.
3. An analysis of the effect of automatic feature selection methods applied to DDoS detection.
4. A comparison of unsupervised, supervised, and semi-supervised algorithms, analyzing their limitations and suggesting in which cases they are more adequate.

Organization. The remaining of this paper is structured as follows. Section 2 presents the background in DDoS and in outlier detection algorithms. Section 3 reviews some works on DDoS detection with machine learning. Section 4 details our dataset, and our methodology for algorithms evaluation. Section 5 presents our results and, finally, Section 6 shows our learnings, conclusions, and suggestions for future works.

2. Background

In this section, we briefly explore the research interest attack, DDoS. We also present the definitions and main ideas of the evaluated algorithms.

Types of DDoS Attack. DDoS are classified into three categories namely application layer attacks, state-exhausting attacks, and volumetric attacks [He et al. 2017]. Known as the low volumetric attack, Application layer ones are the most stealth, may come from one or more machines, and usually are the hardest to identify. Its behavior works with a low traffic rate sent to the victim. Slowloris and Slow HTTP POST are examples of this DDoS category.

Otherwise, state-exhausting and volumetric attacks exploit the use of server resources and network bandwidth. On the one hand, state-exhausting attack aims to consume all networks or bandwidth resources, such as in Ping of Death, an example that causes buffer overflow on the datalink layer with malformed pings. On the other hand, volumetric attack interrupts both network resources and bandwidth to deny benign services as in Memcached Amplification and in DNS Amplification wherein the amplification is cached-used to multiply the attack with IP spoofing to hide the attacker.

Outlier Detection Algorithms. Outlier detection algorithms may be unsupervised, supervised, or semi-supervised novelty detection techniques. Supervised algorithms learn model representation from a labeled dataset. In contrast, unsupervised algorithms rely on pre-defined characteristics of outliers, not using labels for this process. Finally, semi-supervised novelty detection algorithms are trained only with normal behavior points. New points are then considered normal only if they follow a similar distribution to the model. The algorithms used in this work are further described, from unsupervised to semi-supervised and finally supervised ones.

Isolation Forest [Liu et al. 2008], also known as iForest, is a tree-based unsupervised algorithm. IForest considers that unusual instances, outliers, usually are isolated on

the top nodes of a decision tree, i.e. near to the root. The path length to a given point is then used to calculate the outlier score of this element. Iforest is also developed to be an efficient algorithm, with linear time complexity, so it's expected to perform well on large datasets.

HBOS [Goldstein and Dengel 2012], in turn, is a histogram-based unsupervised method. For each given feature, HBOS creates a histogram capable of dealing with categorical or numerical features. For categorical features, the number of occurrences is counted and used as bin height. For numerical, observations are counted according to bin width. Bin densities are then used to calculate outlier scores. The algorithm is specially indicated for network security problems, given its efficiency for large datasets.

PCA and Auto-encoders are two techniques that can be used for semi-supervised outlier detection. On the one hand, PCA [Shyu et al. 2003] is a linear dimensionality reduction algorithm that projects data into a lower-dimensional space. A lower-dimensionality space is argued to be able to turn outlier characteristics explicit, while considering features together in a multivariate approach. For outlier detection, the algorithm uses the point distance to the center of the modeled data to calculate outlier scores. On the other hand, auto-encoders [Sakurada and Yairi 2014] are neural networks also capable of performing dimensionality reduction, however, able to capture nonlinear dependence among variables. In this case, the reconstruction error is used as outlier score.

XGBOD [Zhao and Hryniewicki 2018], in turn, is a supervised ensemble method that uses the scores from unsupervised algorithms as new appended features. XGBOD authors argue that those scores may represent valuable knowledge for outlier classification. New features may then join original ones in an attempt to improve the results. XGBOD relies on XGBoost, another supervised algorithm, for final classification.

Finally, XGBoost [Chen and Guestrin 2016] is a tree-based method that can be used for classification problems. XGBoost implementation includes parallel learning to improve efficiency and also a regularization term to avoid overfitting. Unlike other described algorithms, XGBoost was not specifically designed for outlier detection tasks, however, it can be used to model data distribution and to predict the labels of new instances, turning outlier detection into a binary-classification traditional task. Furthermore, XGBoost implementation is able to generate feature importance, e.g. by counting tree split times over a feature. XGBoost is not implemented by PyOD library.

3. Related Work

In this section, we review some of the recent works on DDoS identification using machine learning. We also present a paper evaluating algorithms for outlier detection on general tasks.

Niyaz et al. [Niyaz et al. 2017] propose a deep learning approach for DDoS detection in software-defined networking. Their supervised model achieves over 99% f1-score on binary classification. Also with a supervised approach, Prasad and PBV [Prasad and PBV 2019] use XGBoost as a classifier for DDoS and benign flow. They aggregate data flow from three other datasets to create their own and achieve 100% f1-score on the binary classification. The dataset used in our paper is a subset of theirs, as described in Section 4.

With an unsupervised approach, Goldstein and Uchida [Goldstein and Uchida 2016] evaluate 19 algorithms for different anomaly detection tasks, including network attacks identification on KDD99 dataset, which covers DoS and DDoS attacks. They, however, kept only 500 instances of these attacks (0.08%), in an attempt to prevent large clusters among malign traffic. HBOS outperforms all algorithms both in AUC and computer efficiency for this dataset, achieving 99.9% AUC in 3.6 seconds.

Still evaluating KDD intrusions, Laskov et al. [Laskov et al. 2005] present a slightly broader view comparing the performance of two approaches for outlier detection. In this case, the authors notice that using high levels of samples, supervised algorithms had better results against unsupervised ones based on the ROC analysis. Also implementing supervised and unsupervised learning, He et al. [He et al. 2017] achieve high F1-Score levels, around 87% and 99% respectively, using four subtypes of DDoS attack to build their dataset.

4. Methodology

In this section, we describe our approaches for data handling and feature selection. We also specify the criteria used for evaluating the proposed algorithms.

Data description and selected sets. Data obtained from Prasad and PBV [Prasad and PBV 2019] is a union of synthetic datasets from the Canadian Institute for Cybersecurity (CIC) Canada[3] extracting only DoS and DDoS attacks, and benign flows. The final dataset is composed of 83 features, 7 being descriptive and the others being quantitative variables. Original data includes a training balanced dataset and a testing unbalanced dataset. For the proposed analysis with unsupervised algorithms, we use a subset of the unbalanced dataset, composed of 60,000 observations. For supervised algorithms, we also work with the subset of the unbalanced data, using the subset of 60,000 as training and another subset with 20,000 for test. Finally, for semi-supervised novelty detection, we use the same sets of the supervised task, however, we remove the DDoS examples from the training one, remaining 49,000 benign samples. Subsets are randomly obtained from the original data.

Automated feature selection. For comparison, we run unsupervised and semi-supervised algorithms in two situations: with and without feature selection. Feature selection is supervised, performed using XGBoost. We select features that split at least three nodes on XGBoost trees, remaining 17 variables, shown in Figure 1. With this process, the idea is not to turn the unsupervised and semi-supervised algorithms into supervised ones, but to illustrate a situation where only important features are passed to the algorithms. A similar result for feature selection could also be obtained from the already existing literature or by expert knowledge. It's expected for those algorithms to perform better with feature selection, as original features might include a lot of noise. The supervised algorithms evaluated already perform this feature selection internally, therefore, automated feature selection for them is not externally done as it is in the others.

Manual feature selection. The original dataset also contains 8 features that always assume the value 0. Those features are already removed from all sets to reduce data

[3]CIC Canada datasets: https://www.unb.ca/cic/datasets/. Accessed in 15 jul 2020.

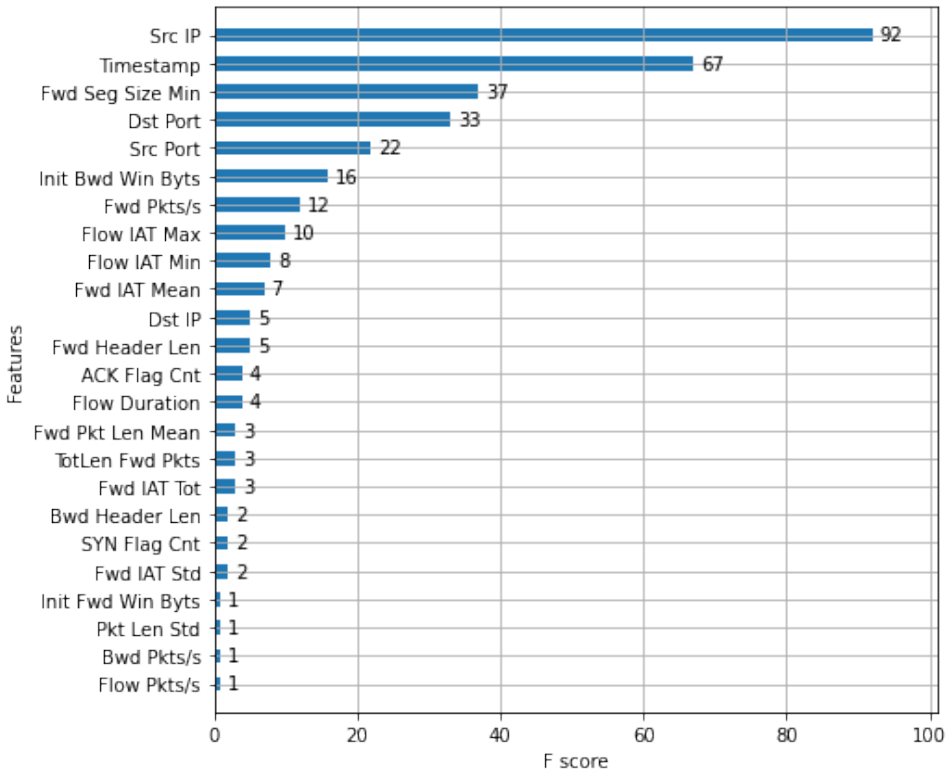

Figure 1. XGBoost feature importance using F score: number of times a variable splits nodes on XGBoost tree

dimension. Furthermore, a feature called "Flow ID", represented with high importance by Prasad and PBV [Prasad and PBV 2019] is also dropped, as we notice it is composed by a direct union of other variables: "Src IP","Dst IP","Src Port","Dst Port", and "Protocol". This explains its high importance for XGBoost, however, indicates a distortion on feature balance for other methods, as those features would appear two times in data. An "unknown" feature is also removed as we found no information about it. Therefore, the remaining features are 74.

Algorithms evaluation. Algorithms are compared focusing on F1-score [3] and Area Under the Receiver Operating Characteristic Curve (AUC). F1-score provides a balance between precision [1] and recall [2], while AUC allows considering true positive (TP) and false positive (FP) rates face to face. For precision, recall, and F1, we use the expected proportion of DDoS on our datasets as threshold for considering an observation an outlier. We also compare the efficiency of the algorithms using the total amount of time for fitting and predicting. Specifically for XGBOD, we use HBOS as the ensembled algorithm. As it is based on XGBoost, for that one, we also show the feature importance model after ensemble, in order to illustrate the difference in the decision process. Tests are performed in a server with Intel Core i9-9900X and 125GB RAM.

$$Precision = \frac{TP}{TP + FP} \quad (1) \qquad Recall = \frac{TP}{TP + FalseNegatives} \quad (2)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

5. Results and Discussion

In this section, we show the obtained results while evaluating the performance of the proposed algorithms. Table 1 shows the final results of the performed tests. Both supervised algorithms achieve perfect results even working with subsets. In unsupervised ones, HBOS stands out after feature selection and is the fastest algorithm. We also notice that feature selection makes a big difference especially for it. In terms of execution time, HBOS also outperforms all other methods. IForest, in turn, relies on the usually few amount and different characteristics of outliers, which is not always a fact when dealing with DDoS, as discussed in Section 2. This characteristic might explain its performance.

On semi-supervised ones, PCA and auto-encoder have very similar results and present slightly worse F1 results after feature selection on the defined threshold. This might indicate that their multivariate analysis and feature reduction already are capable of promoting a similar result of feature selection. AUC, however, presents a considerably better performance, indicating that this is not necessarily a fact for all thresholds. Even though PCA is unable to capture nonlinear correlations, it is capable of getting basically the same results of auto-encoder. Both present less time for execution, but PCA is significantly smaller in contrast to auto-encoder, which doesn't change so much.

Table 1. Results. "w/fs": "with feature selection".

Algorithm	Type	Precision	Recall	F1	AUC	CPU time (s)
iForest	Unsupervised	0.13	0.13	0.13	0.57	12.63
iForest w/fs	Unsupervised	0.22	0.22	0.22	0.70	5.64
HBOS	Unsupervised	0.18	0.18	0.18	0.67	0.31
HBOS w/fs	Unsupervised	0.71	0.71	0.71	0.89	0.07
PCA	Semi-supervised	0.37	0.46	0.41	0.68	0.54
PCA w/fs	Semi-supervised	0.36	0.46	0.40	0.81	0.11
Auto-encoder	Semi-supervised	0.37	0.46	0.41	0.69	26.32
Auto-encoder w/fs	Semi-supervised	0.36	0.46	0.40	0.82	26.26
XGBoost	Supervised	1.00	1.00	1.00	1.00	13.88
XGBOD	Supervised	1.00	1.00	1.00	1.00	16.50

Figure 2 shows the feature importance model on XGBOD ensembled with HBOS. HBOS derivated feature does not receive high importance, however its presence is remarkably capable of influence the final tree of XGBoost. It is comprehensible that it doesn't receive a big feature importance, given its bad performance without feature selection. However, even with a different tree, the result keeps perfect.

6. Conclusion

In this paper, we evaluated some unsupervised, semi-supervised, and supervised algorithms for outlier detection. We also analyzed the effect of feature selection on each of them. We've shown that the tested supervised algorithms fit perfectly for our data, even though we've worked only with subsets of the original dataset. Therefore, we illustrated that even a smaller amount of data can generalize the characteristics of DDoS attacks when working with supervised methods.

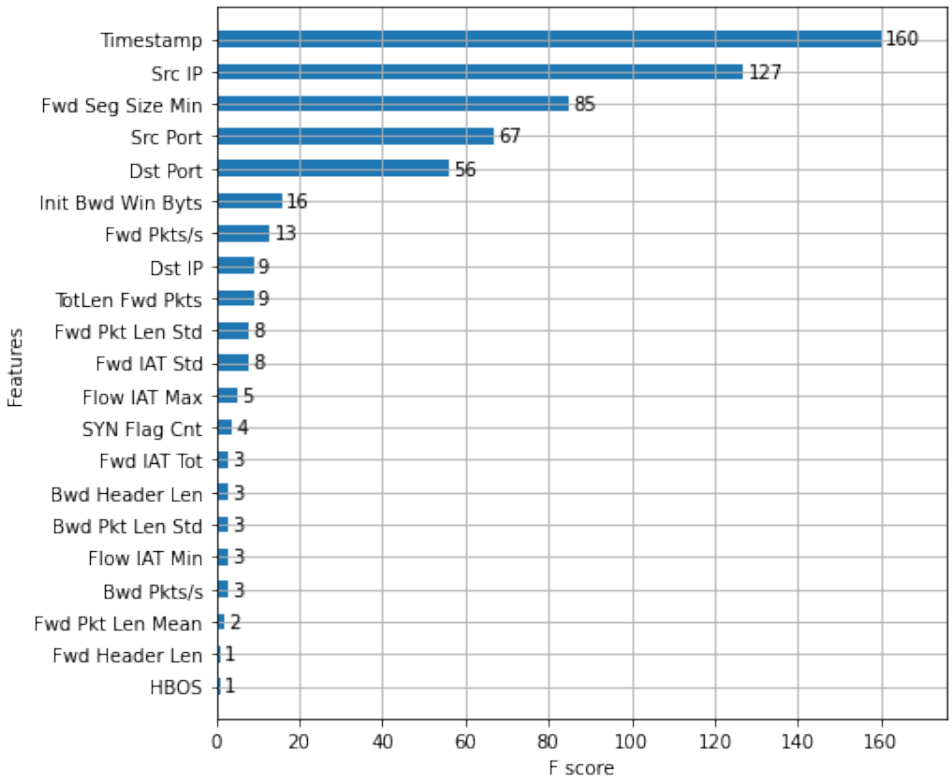

Figure 2. XGBoost feature importance on XGBOD final tree.

In our case, unsupervised and semi-supervised algorithms do not perform so well for DDoS detection, however, they are usually faster and do not rely on labeled data. Particularly, HBOS results indicated better performance after feature selection, illustrating that selecting the right features is really important for DDoS identification, especially for unsupervised algorithms. We also notice that the generated scores of those algorithms are able to considerably influence the decision process of XGBoost, through the method proposed by the XGBOD [Zhao and Hryniewicki 2018]. Therefore, ensembling those algorithms might be a good idea for DDoS identification datasets that, even with supervised approaches, do not get perfect results right away.

We hypothesize that the worst results of those unsupervised and semi-supervised algorithms are related to the nature of DDoS attacks, which includes their distributed characteristic, usually big proportion, and their possibility to constitute clusters that might be confused with normal behavior data. For future works, we suggest analyzing the effect of the proportion of DDoS observations on those datasets and also evaluating other unsupervised and semi-supervised algorithms. Another further analysis would be removing the Src IP and the Timestamp features. Those features might be obvious for well-defined attacks in supervised algorithms, as the attacks may come from specific sources in specific times.

7. Acknowledgments

The authors would like to thank the Pró-Reitoria de Pesquisa (PRPq/UFMG),the Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), and the Con-

selho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for the support.

References

[Chen and Guestrin 2016] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

[Goldstein and Dengel 2012] Goldstein, M. and Dengel, A. (2012). Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm.

[Goldstein and Uchida 2016] Goldstein, M. and Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173.

[He et al. 2017] He, Z., Zhang, T., and Lee, R. B. (2017). Machine learning based ddos attack detection from source side in cloud. In *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*, pages 114–120. IEEE.

[Laskov et al. 2005] Laskov, P., Düssel, P., Schäfer, C., and Rieck, K. (2005). Learning intrusion detection: supervised or unsupervised? In *International Conference on Image Analysis and Processing*, pages 50–57. Springer.

[Liu et al. 2008] Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE.

[Niyaz et al. 2017] Niyaz, Q., Sun, W., and Javaid, A. Y. (2017). A deep learning based ddos detection system in software-defined networking (sdn). *ICST Transactions on Security and Safety*, 4(12):153515.

[Prasad and PBV 2019] Prasad, M. D. and PBV, C. A. (2019). Machine learning ddos detection using stochastic gradient boosting. *International Journal of Computer Sciences and Engineering*, 7(4):157–16.

[Sakurada and Yairi 2014] Sakurada, M. and Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11.

[Shyu et al. 2003] Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., and Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. Technical report, MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING.

[Sonar and Upadhyay 2014] Sonar, K. and Upadhyay, H. (2014). A survey: Ddos attack on internet of things. *International Journal of Engineering Research and Development*, 10(11):58–63.

[Zhao and Hryniewicki 2018] Zhao, Y. and Hryniewicki, M. K. (2018). Xgbod: improving supervised outlier detection with unsupervised representation learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[Zhao et al. 2019] Zhao, Y., Nasrullah, Z., and Li, Z. (2019). Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20:1–7.