Benchmarking bioinspired machine learning algorithms with CSE-CIC-IDS2018 network intrusions dataset

Paulo Ferreira[1]
2180047@my.ipleiria.pt
Mário Antunes[123]
mario.antunes@ipleiria.pt

[1] School of Technology and Management
Polytechnic of Leiria - Portugal
[2] CIIC, Computer Science and Communication Research Centre, ESTG, Polytechnic of Leiria - Portugal
[3] Center for Research in Advanced Computing Systems INESC-TEC, University of Porto - Portugal

Abstract

This paper aims to evaluate CSE-CIC-IDS2018 network intrusions dataset and benchmark a set of supervised bioinspired machine learning algorithms, namely CLONALG Artificial Immune System, Learning Vector Quantization (LVQ) and Back-Propagation Multi-Layer Perceptron (MLP). The results obtained were also compared with an ensemble strategy based on a majority voting algorithm. The results obtained show the appropriateness of using the dataset to test behaviour based network intrusion detection algorithms and the efficiency of MLP algorithm to detect zero-day attacks, when comparing with CLONALG and LVQ.

1 Introduction

Computer networks security encloses a wide set of technologies to protect the assets and the users operation. Due to its operating mode, Intrusion Detection System (IDS), namely those based on behaviour analysis, are able to detect, with some degree of accuracy and in a timely manner, zero-day attacks and vulnerabilities exploits, to further apply countermeasures. In this paper we intend to evaluate a set of bioinspired algorithms already developed and implemented by Machine Learning (ML) tools. The major contributions can be summarized as follow: i) an open source framework and processing flow, based on WEKA [1], to ingest and process CSE-CIC-IDS2018 dataset; ii) an open source tool to automate the tests carried on with CLONALG [2], LVQ [3] and Backpropagation-MLP [4] classifiers; iii) a comparison between the results obtained individually by each of the bioinspired algorithms with those achieved by an ensemble approach with the same models, using *majority voting* strategy. This paper is organized as follows: Section 2 describes the key concepts for this work. The tests setup is described in section 3, the results are presented in section 4 and further analysed in section 5. Conclusions and future work are described in section 6.

2 Background

IDS can be classified according to the object of analysis (host-based or network-based) and according to the detection method (behaviour-based or signature-based). Behaviour-based IDS aim to overcome the limitations observed on those that are signature-based, namely its inability to detect patterns that are not in a predefined signature database. These systems analyse traffic and try to define a normal network behaviour to further identify deviations that are considered anomalous traffic and, therefore, reported as possible positive examples [5].

Bioinspired ML algorithms are a set of algorithms whose operation is mimicked on systems or mechanisms from the nature or the human body. Some typical applications and analogies are the neural networks, inspired by the functioning of the human brain; the evolutionary and DNA computing, based on theories of evolution that leads to genetic algorithms; the Artificial Immune Systems (AIS), which takes inspiration on the vertebrate immune system, namely its adaptive part [6]. Regarding Artificial Neural Networks (ANN) algorithms, in this work we have used Back-propagation Multi-Layer Perceptron (MLP) [4] and Learning Vector Quantization (LVQ) [3]. From the whole plethora of immune-inspired algorithms [7], the one chosen for this work was CLONal selection ALGorithm (CLONALG) [2].

The tests were carried out with the CSE-CIC-IDS2018 public dataset[1]. Despite being recent, CSE-CIC-IDS2018 dataset is very well organized and is now starting to be widely used by the scientific community to benchmark IDS. It includes a wide range of attacks, executed with different tools, organized in a timeline and mixing both normal and anomalous network packet flows. The traffic was dynamically generated, with the purpose of simulating a corporate network.

Due to the wide variety of attacks and the deluge of data available, we have defined a subset of attacks that could better test the detection of a previously unseen attack. The choice was also based on the diversity and amount of data related to each attack. Table 1 describes the characterization of the attacks used in the experiments carried on in this paper.

Table 1: Network attacks characterization

Date	Time		Type of attack	Software Tool	# flows
	Begin	End			
16/02/2018	10:12	11:08	DoS	SlowHTTPTest	139890
	13:45	14:19	DoS	Hulk	461912
21/02/2018	10:09	10:43	DDoS	LOIC-UDP	1730
	14:05	15:05	DDoS	HOIC	686012

The number of normal traffic flows available at each date is 446772 and 360833 respectively for 16/02/2018 and 21/02/2018.

3 Tests setup

We have carried out four test scenarios, as can be seen on table 2.

Table 2: Test scenarios

Scenario	Training		Testing	
	Date	Traffic	Date	Traffic
1	16/02/2018	Normal+Attack1	16/02/2018	Normal+Attack2
2	16/02/2018	Normal+Attack1	21/02/2018	Normal+Attack1
3	16/02/2018	Normal+Attack2	21/02/2018	Normal+Attack2
4	16/02/2018	Normal+Attacks	21/02/2018	Normal+Attacks

The tests were performed on a subset with 200,000 instances, that is network flows. From that value, 70% (140,000 records) of them constitute the training dataset and the remaining 30% (60,000 records) are part of the testing dataset. The training set records are selected from the training data file and the test set records are selected from the test data file. Each test scenario was then run ten times, with independent data for each iteration, but the same for the three algorithms in each iteration.

Besides the three algorithms mentioned above, we have also considered an *ensemble* of the models generated by the three algorithms, in which the decision strategy is based on the criterion for majority decision, also known as *majority voting*.

The methodology used to run the experiments consists of four main phases: input data ingestion, data preprocessing, data processing and presentation of results (see figure 1).

Figure 1: Methodology

The preprocessing phase deals with issues like removing unnecessary attributes, normalizing data, reducing the number of classes by aggregating every class not being "Benign" as malicious traffic and dealing with missing values by replacing them with the average value for each attribute. These tasks were essentially accomplished through WEKA [1] and Orange [8] *open-source* applications.

[1] https://registry.opendata.aws/cse-cic-ids2018/

The preprocessed dataset is then processed by the algorithms in both training and testing phases. We have used WEKA for that purpose and have also developed an application to automate the tests for any dataset that meets the requirements[2].

4 Results

Tables 3 through 6 show the results obtained for each of the scenarios listed in Table 2. For a given algorithm, the values of each metric correspond to the arithmetic mean of the values obtained for all the ten iterations.

Table 3: Results for scenario 1

Algorithm	TPR	TNR	FPR	FNR	Precision	Recall	Accuracy	F1
CLONALG	0,0306	0,9997	0,0003	0,9694	0,9895	0,0306	0,5071	0,0593
LVQ	0,0306	0,9996	0,0004	0,9694	0,9889	0,0306	0,5071	0,0593
MLP	0,0001	1,0000	0,0000	0,9999	1,0000	0,0001	0,4917	0,0001
Ensemble	0,0306	0,9997	0,0003	0,9694	0,9895	0,0306	0,5071	0,0593

Table 4: Results for Scenario 2

Algorithm	TPR	TNR	FPR	FNR	Precision	Recall	Accuracy	F1
CLONALG	0,0080	0,6537	0,3463	0,9920	0,0337	0,0080	0,6506	0,0103
LVQ	0,7025	0,0031	0,9969	0,2976	0,0034	0,7025	0,0065	0,0067
MLP	0,0000	0,9998	0,0003	1,0000	0,0000	0,0000	0,9950	0,0000
Ensemble	0,0080	0,6537	0,3463	0,9920	0,0337	0,0080	0,6506	0,0103

Table 5: Results Scenario 3

Algorithm	TPR	TNR	FPR	FNR	Precision	Recall	Accuracy	F1
MLP	1,0000	0,9998	0,0002	0,0000	0,9999	1,0000	0,9999	0,9999
CLONALG	1,0000	0,0026	0,9974	0,0000	0,6559	1,0000	0,6562	0,7922
LVQ	1,0000	0,0004	0,9997	0,0000	0,6554	1,0000	0,6554	0,7918
Ensemble	1,0000	0,0026	0,9974	0,0000	0,6559	1,0000	0,6562	0,7922

Table 6: Results for Scenario 4

Algorithm	TPR	TNR	FPR	FNR	Precision	Recall	Accuracy	F1
MLP	0,8977	0,9996	0,0004	0,1023	0,9284	0,8977	0,9327	0,8987
LVQ	1,0000	0,0008	0,9992	0,0000	0,6560	1,0000	0,6561	0,7923
CLONALG	0,9992	0,0030	0,9970	0,0008	0,6564	0,9992	0,6564	0,7923
Ensemble	0,9992	0,0033	0,9968	0,0008	0,6564	0,9992	0,6565	0,7923

5 Results Analysis

The purpose of the tests was to simulate the detection of a *zero-day* attack, by using the CSE-CIC-IDS2018 dataset. It is appropriate to mention that a network attack is essentially an anomaly to the normal network traffic behaviour. It may be seen, for example, as a high traffic volume in a short period of time, so it might be important to identify the parameters that allow the system to detect these examples. Some of these parameters could be the number of packages per time interval or the time interval between each package.

Regarding the ensemble classifier, as we can see in the results, given that two of the three classifiers always present very unfavorable results, the contribution of the *ensemble*, if any, is not significant.

The results are promising in some way, as the tools used in the attacks have produced patterns with some resemblance, thus making it possible for a behaviour-based IDS to use these algorithms to be able to identify a *zero-day* attack.

In scenarios 1 and 2, despite having a low True Positive Rate (TPR), the CLONALG algorithm stands out, together with the *ensemble*, as can be seen from the F1 values. The MLP algorithm has shown to be incapable of handling with this kind of traffic, only correctly identifying the overwhelming majority of normal traffic.

In contrast, in scenario 2, the LVQ algorithm presented the highest TPR in the scenario, despite failing to identify normal traffic (lowest True Negative Rate (TNR) value in the scenario).

In scenarios 3 and 4, we can depict the predominance of the MLP algorithm, with high F1 values, very close to 100% in scenario 3.

In scenario 3, as can be seen in the table 5, all algorithms correctly identified all malicious traffic (TPR = 1), which may be related to the similarity of traffic patterns generated by the respective tools. With regard to normal traffic, only MLP performs well, with TNR very close to 100%, while the other algorithms have a very residual identification.

In scenario 4, despite the great diversity of malicious traffic both in the training and testing phases, the traffic generated by the two tools in each type of attack has no significant advantage when compared to the results obtained in scenario 3. In fact, the performance of MLP, translated by the F1 value, drops by about 10%, whereas, in the other algorithms, there is little improvement.

6 Conclusions and Future Work

In this paper we have described a methodology to test bioinspired machine learning algorithms, against the recent and promising CSE-CIC IDS-2018 dataset. We described the dataset and the methodology used to process the four scenarios defined in each module. To fully automate the tests we have made available a tool developed with WEKA Java API.

We have sought to obtain statistical significance by running the tests ten times for each algorithm. The parameters set used in each algorithm was obtained empirically, combining the requirements of the algorithm itself and the data to be analysed.

In the first two scenarios, the highlighted algorithm is CLONALG, although the TPR is quite low, while MLP algorithm reveals poor performance. Despite correctly identifying the overwhelming majority of normal traffic, it clearly fails to identify malicious traffic. In the scenarios 3 and 4, the MLP performance is promising, with F1 and TPR values above 89%.

In addition to results obtained by each algorithm individually, an *ensemble* classifier was also implemented, which, using a majority voting strategy, had no significant influence in the final results. The future work includes the optimization of the parameters set and the processing of others datasets derived from CSE-CIC IDS-2018 dataset, with different attacks for training and testing.

References

[1] E. Frank, M. A. Hall, and I. H. Witten, "The weka workbench," in *Data Mining: Practical Machine Learning Tools and Techniques*, M. Kaufmann, Ed., 4th ed. 2016, ch. Online Appendix. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf.

[2] L. N. de Castro and F. J. Von Zuben, "The clonal selection algorithm with engineering applications," in *Proceedings of GECCO*, editor, Ed., 2000, pp. 36–39.

[3] T. Kohonen, *Self-Organizing Maps*, ser. Springer Series in Information Sciences. Springer Science & Business Media, 2001, vol. 30. DOI: 10.1007/978-3-642-56927-2.

[4] F. Amato, N. Mazzocca, F. Moscato, and E. Vivenzio, "Multilayer perceptron: An intelligent model for classification and intrusion detection," in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, IEEE, 2017, pp. 686–691.

[5] A. Fuchsberger, "Intrusion detection systems and intrusion prevention systems," *Information Security Technical Report*, vol. 10, pp. 134–139, 3 2005. DOI: 10.16/j.istr.2005.08.001.

[6] M. Mahboubian and N. A. W. A. Hamid, "A machine learning based ais ids," *International Journal of Machine Learning and Computing*, vol. 3, no. 3, pp. 259–262, Jun. 2013. DOI: 10.7763/IJMLC.2013.V3.315.

[7] J. Kim, P. J. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, "Immune system approaches to intrusion detection–a review," *Natural computing*, vol. 6, no. 4, pp. 413–466, 2007.

[8] J. Demšar, T. Curk, A. Erjavec, Č. Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan, "Orange: Data mining toolbox in python," *Journal of Machine Learning Research*, vol. 14, pp. 2349–2353, 2013.

[2] https://github.com/paulo-ferreira-mcif/benchmarkids