Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 196 (2022) 133–141

Procedia Computer Science

www.elsevier.com/locate/procedia

CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021

Exploring Dataset Manipulation via Machine Learning for Botnet Traffic

Rodrigo Abrantes[a], Pedro Mestre[a,b], António Cunha[a,c,*]

[a]*Universidade de Trás-os-Montes e Alto Douro, Vila Real, 5000-801, Portugal*
[b]*Centro Algoritmi, Universidade do Minho, Guimarães 4800-058, Portugal*
[c]*INESC TEC – INESC Technology and Science, Porto 4200-465, Portugal*

Abstract

Botnets are responsible for some of the major malicious traffic on the Internet: DDoS attacks, Mail SPAM, brute force attacks, portscans, and others. Its dangerousness is due to the coordinated amount of infected hosts focusing on a single target. More contributions are in need, considering that (A) ML has been used for cyberattacks identification with better accuracy than standard NIDS equipments, (B) Botnet attacks are one of the most dangerous threats on the Internet. (C) the difficulties in getting representative datasets on some Botnets, and (D) Botnet traffic can be misunderstood by its infrastructure protocol.

In this paper, we focus on the identification of Botnet traffic, preventing the communication from the Botmaster to the infected hosts and consequently the Botnet cyberattacks. CICFlowMeter and Machine Learning algorithms were used to analyse Botnet2014 public dataset on four different scenarios: all Botnet traffic on a single class, each class per Botnet traffic and the influence of the IPs address fields Botnet traffic detection.

The results shows that Random Forest (RF) and Decision Tree (CART) archived similar accuracies on Botnet traffic classification. Important to say that CART obtained similar results with 10-20% of machine time. The metrics shown that the analysis per specific Botnet has higher accuracy than Any Botnet Traffic analysis. Also, the analysis with the IP addresses and L4 Ports scenario has higher accuracy but lower F1-Score that the equivalent without IP addresses or L4 Ports. At last, Feature Importance results confirms the literature, that Botnet traffic is not a single uniform protocol, but a collection of very different ways of communications between the botmaster and the infected hosts.

* Corresponding author.
E-mail address: acunha@utad.pt

1877-0509
10.1016/j.procs.2021.11.082

Keywords: CICFlowMeter, Botnet2014, Botnet Traffic, Machine Learning, Random Forest Classifier, Decision Tree Classifier.

1. Introduction

ARPANET emerged in 1969 from the interconnection of military networks and has since evolved into today's INTERNET by aggregating academic and commercial networks. In this scenario, several types of equipments for monitorization, control, and defence are used as protection against cyberattacks, highlighting firewalls, Intrusion Detection System (IDS) and Intrusion Prevention System (IPS). IDS can be subdivided into a network (NIDS) or host (HIDS), which will not be the target of this research. In an attempt to improve the accuracy of the classification of cyberattacks, Artificial Intelligence (AI) mechanisms have been used to analyse attack patterns and network traffic [6, 21, 27, 29]. It might also provide mechanisms for detecting day-zero attacks [29].

Botnets are relevant for the cybersecurity area because they can build a network of infected devices to orchestrate one of the major malicious traffic on Internet [13, 25, 2]: DDoS attacks, Mail SPAM, brute force attacks, portscans, and others. It is important to say that Botnet traffic is not the attack itself but the backhaul of communication from the botmaster to the infected hosts (aka bots). Its dangerousness is due to the coordination of the number of infected hosts focusing on a single target. Botnet traffic was also chosen by its diversity. Literature shows very different ways of communication, using standard protocols as its infrastructure of communication (such as P2P, IRC, Web and SSH [19, 26]). This behaviour makes more complex Botnet traffic identification due to they can be misunderstood by the infrastructure protocol itself. This is an ongoing field of study. It can be found in literature at least 633 papers in the last five years from WoS site regarding Botnet.

More contributions are in need, considering that (A) ML has been used for cyberattacks identification with better accuracy than standard NIDS equipments, (B) Botnet attacks are one of the most dangerous threats on the Internet. (C) the difficulties in getting representative datasets on some Botnets, and (D) Botnet traffic can be misunderstood by its infrastructure protocol.

In this paper, we focus on the identification of Botnet traffic, preventing the communication from the Botmaster to the infected hosts and consequently the Botnet attacks. Important to say that this is an initial work, part of a bigger research project that aims the development of explainable methods for detecting cyberattacks. We will discuss the stages of the dataset manipulation and the ML algorithms, the findings and challenges faced during this analysis and the next steps of this research.

2. Related Work

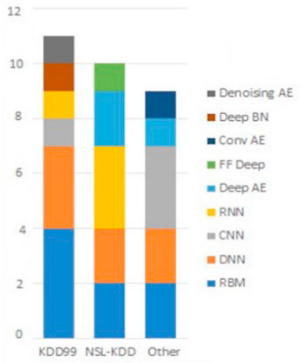
Below we will see related work in the field of Artificial Intelligence for detecting Botnet traffic. Also, since one of the first public datasets related to cybersecurity, several others were created for specific usages in this area. The ML applications classifies the public datasets for better cyberattacks detection than the standard IDS.

2.1. Public datasets

In 1999, The Fifth International Conference on Knowledge Discovery and Data Mining held a competition to analyse the proposed dataset in search of cyberattacks, emerging the KDD99 Dataset [14]. From this point, others were created for use in the academic and commercial fields. [15] compiled a detailed listing of 34 datasets from 1998 to 2018, where we can see an example of this diversity. This benchmark can also be observed in Fig. 1 a). According to [6], on Figure 1 we can see that 20 of 30 researches used KDD99 and NSL-KDD datasets. Besides its popularity, we can also see that these datasets are experimented with a variety of AI algorithms, helping in future comparison with others researches. Some datasets such as Botnet2014 [3], Tor-nonTor [12] or BoT-IoT [11] are related to Botnet traffic and can also be used in researches in this specific fields. Some of the Botnet traffic are not well represented on public datasets, as we can see in Table 2.

2.2. Machine learning applications

On the analysis of cyberattacks by AI, we can see very different traffic classification rates just changing the AI algorithm, for the same kind of attack, on the same dataset. The accurate identification of the cyberattack is highly dependent on the used AI Algorithm, as calculated by [29, 6, 21]. Besides that, some algorithms are well performed on most of the cyberattacks, such as: MLP, RF, RNN and CNN [21, 4, 22]. An example of the differences per cyberattack by AI Algorithms [29] is shown in Fig. 2.



a) AI Methods by Dataset

Botnet name	Type	Portion of flows in data-set
Neris	IRC	25967 (5.67%)
Rbot	IRC	83 (0.018%)
Menti	IRC	2878(0.62%)
Sogou	HTTP	89 (0.019%)
Murlo	IRC	4881 (1.06%)
Virut	HTTP	58576 (12.80%)
NSIS	P2P	757 (0.165%)
Zeus	P2P	502 (0.109%)
SMTP Spam	P2P	21633 (4.72%)
UDP Storm	P2P	44062 (9.63%)
Tbot	IRC	1296 (0.283%)
Zero Access	P2P	1011 (0.221%)
Weasel	P2P	42313 (9.25%)
Smoke Bot	P2P	78 (0.017%)
Zeus Control (C&C)	P2P	31 (0.006%)

b) Botnet2014 test set

Botnet name	Type	Portion of flows in data-set
Neris	IRC	21159 (12%)
Rbot	IRC	39316 (22%)
Virut	HTTP	1638 (0.94 %)
NSIS	P2P	4336 (2.48%)
SMTP Spam	P2P	11296 (6.48%)
Zeus	P2P	31 (0.01%)
Zeus control (C & C)	P2P	20 (0.01%)

c) Botnet2014 training set

Fig. 1: AI methods by dataset (a) and dataset Botnet2014 (b,c) used in this work.

IP flows [20] are the communication from the source to the destination, and they are composed by each one of the packets of the communication. Some flows last few seconds, other days. For identifying it, must be used the source and destination IP addresses and Ports as well, as the Layer 4 Protocol (TCP or UDP). As said by [23]: "Training Classifiers (…) can cause overfitting problems and induce them to learn to distinguish malicious flows only on the basis of IP addresses and/or port numbers".

The author of the Botnet2014 had similar problems when analysing his dataset [3]. The paper got inconclusive findings when using the packet header data (Source and Destination IP addresses and Ports). Important to say that, nevertheless, the usage of this information might be controversial, the source and destination IP addresses and Ports, and the Protocol are essential for identifying the IP Flow.

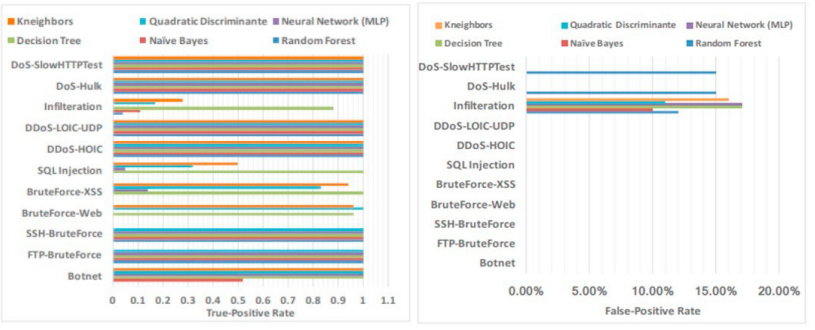


Fig. 2: True-Positive Rate and False-Positive rate achieved by AI Algorithm by Cyberattack. [29]

3. Methods and Materials

The pipeline present in Fig. 3 shows the different phases executed during this work. The public dataset Botnet2014[3] is a pcap based dataset, subdivided into 3 different datasets. It was manipulated, aiming to evaluate different state-of-the-art architectures. The application CICFlowMeter [18, 5] identified the IP Flows [20] from the

pcap files and statistical information based on the IP Flows were calculated, generating csv files. These csv files were merged and its duplicated records were removed. The generated datasets were merged on a single datafile (with 648390 records), being splitted into train, validation and test sets, respectivelly with 35%, 35% and 20% of the datafile. The ML analisys of the statistical IP Flow considers two diferent variables: A) analisys of each Botnet traffic and all Botnet traffic; B) analisys with ot without Source and Destinations IP addresses and Ports.

Fig. 3: Overview of the developed work.

3.1. Choose the Dataset

Botnet2014 is a compilation of the datasets: ISOT [28], ISCX 2012 [17] and Malware Capture Facility Project 2012. It is separated on Training (Aprox. 5.3GB) and Testing dataset (Aprox. 8.5GB) and represents traffic from 16 botnets with different types of traffic, in pcap format. The dataset has a traffic categorisation of each Botnet by its Source IP address.

The Botnet2014 dataset [3] was chosen due to the traffic representation on most Botnet traffics. Fig. 2 (b,c) shows the number of Botnet records by database [3]. The pcap format is a dumping of the data that was sent across the network. Each pcap record contains the L2 ethernet frame header and the IP packet header and payload.

3.2. Prepare the Dataset

For identifying the patterns by Botnet traffic, the IP Flows were identified, and for each IP Flow was generated statistical information in csv format. This was done by the application CICFlowMeter [18, 5]. This statistical information will be used for identifying the patterns of each Botnet. The complete listing of the parameter generated by CICFlowMeter can be obtained in [5]. Each csv file was generated from the original pcap file. Approximately 30% of duplicated records were found on all datasets (considering only the statistical information generated by CICFlowMeter) and deduplicated by the GNU UNIX tools commands presented on the Table 1. It was generated the csv database within 648,390 records to be processed by the ML Algorithm. Detailed output can be found in Table 1.

Table 1: Amount of duplicated and deduplicated records on the datasets.

Description	Unix Executed Command	Amount
Duplicated Entries on ISCX_Botnet Trainning and Testing	cat ISCX_Botnet-Training.pcap_Flow.csv ISCX_Botnet-Testing.pcap_Flow.csv | grep -v ^Flow|sort|uniq -d	1895 (0.2%)
Duplicated Entries on ISCX_Botnet Trainning and Testing and testDset	cat ISCX_Botnet-Training.pcap_Flow.csv ISCX_Botnet-Testing.pcap_Flow.csv testDset-with\ iscx.pcap_Flow.csv | grep -v ^Flow|sort|uniq -d|wc -l	279675 (30%)
Total Entries on ISCX_Botnet Trainning and Testing and testDset	cat ISCX_Botnet-Training.pcap_Flow.csv ISCX_Botnet-Testing.pcap_Flow.csv testDset-with\ iscx.pcap_Flow.csv | grep -v ^Flow|sort|uniq -d|wc -l	930133 (100%)
Deduplicated Entries on ISCX_Botnet Trainning and Testing (With Header)	(grep ^Flow ISCX_Botnet-Training.pcap_Flow.csv;cat ISCX_Botnet-Training.pcap_Flow.csv ISCX_Botnet-Testing.pcap_Flow.csv | grep -v ^Flow|sort|uniq)	648391 (70%)

After being processed by CICFlowMeter, some incorrect information was found (Such as negative times, Infinite and NaN data on the columns) on the database. These fields were changed for outliers data, but other fields of the same record were kept intact. The records of the database were labelled according to the Flow IP Field, identifying the Botnet traffic, as detailed on Table 2. For future reference, besides CICFlowMeter, other tools such as Argos [26] and YAF [15, 7, 8] are also used for pcap conversion to netflow [10, 25] or csv formats. CICFlowMeter was chosen because of the statistical analysis.

Two distinct approaches were used for comparison during ML algorithm: "Has ANY Botnet?" or "Has Each Botnet?". The first one collects all of the Botnets on the same field, as the second one has a separated field for each Botnet traffic. Note a small difference (13 records) between the summaries. This is because some of the traffic was originated from one Botnet Host to the other one. Details below on Table 2.

The Source and Destination Ports were categorized according to IANA [9] definition because it is a strong evidence of the privilege that the application has on the host system, also observed on [23]. Despite the original dataset, the Training and Test datasets were merged (A similar approach was made by [23]) and being resplitted in Training, Test and Validation datasets. This was done because some of the Botnets were not present on the original training dataset, as we can see in the Fig. 1 (b,c).

Table 2: Amount of entries on the datafile by Botnet Classification – Elaborated by the Author

Botnet Type	Amount	Perc Dset	Botnet Type	Amount	Perc Dset	Botnet Type	Amount	Perc Dset
Total Entries	648390	100%	Sogou	82	0%	Weasel	85584	13%
Any Botnet Class	391373	60%	Murlo	12948	2%	Zeus	737	0%
Each Botnet Sum´d	391386	60%	Virut	44454	7%	OsxTrojan	27	0%
IRC	149916	23%	BlackHole1	45	0%	ZeroAccess	1828	0%
Neris	45488	7%	BlackHole2	445	0%	SmokeBot	84	0%
RBot	43558	7%	BlackHole3	252	0%			
Menti	4971	1%	TBot	967	0%			

Considering the controversial usage of Source and Destination IP addresses and L4 Ports for ML Algorithms, both scenarios were analyzed. This was intentional for comparing its metrics on the ML Algorithm. In the first scenario, all information from IP Packet Header was kept. The IP addresses were converted to float so they could be manipulated by ML. On the second one, the Source and Destination IP addresses and L4 Ports were removed. In both cases, the original Label and the Timestamp fields were removed, and the Protocol Field was kept, due to it is related to the behavior of the Botnet traffic itself.

Table 3: Evaluation Metrics description and formulas

Item	Description	Formula
Accuracy	Ratio of the correctly recognized records from the entire dataset	$(TP + TN)/(TP + TN + FP + FN)$
Precision	Ratio of the correctly identified attack records to the number of all	$TP/(TP + FP)$
Recall	Ratio of the correctly classified Attack connection records to the total	$TP/(TP + FN)$
False Positive	Ratio of the Normal connection records tagged as attacks to the total	$FP/(FP + TN)$
F1-Score	Harmonic mean of Precision and Recall	$2x(PrecisionxRecall) / (PrecisionRecall) +$

Legend: TP: Legitimate traffic correctly classified as legitimate traffic; TN: cyberattacks correctly classified as cyberattacks; FP: legitimate records wrongly classified as cyberattacks; FN: cyberattacks wrongly classified as legitimate traffic.

3.3. Train the ML algorithm

Were analyzed seven different classifier algorithms: Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Neighbors (KNN), Decision Tree (CART), Support Vector Machine (SVM) and Naive Gauss Bayes (Bayes). The classifier was trained using 35% of the datafile considering four different scenarios: A) analysis of each Botnet traffic and all Botnet traffic; B) analysis with and without Source and Destinations IP addresses and Ports. All experiments were implemented using Linux Debian 10.9 Virtual Machine, using python [1,16] and TensorFlow [6,30] software libraries.

Four metrics have been used to evaluate the models based on the number of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Each metrics and its formula are detailed on the Table 3.

4. Results

4.1. Classifier Algorithm Comparison

Were analysed seven different classifier algorithms using the first 100.000 records of the dataset, being executed for: ANY, IRC and Neris Botnet traffic with or without IP addresses or L4 Ports. Using or not IP addresses and L4 Ports as part of the dataset, RF (Random Forest) and CART (Decision Tree) gained the best accuracies, with fewer machine time from CART. The comparison between the classifiers and the scenarios are detailed on Table 4. The Selected algorithm was RF because of better overall performance described on the literature for this algorithm. During the runnings, we obtained similar accuracies between RF and CART. CART run time was 10-20% of the RF classifier.

Table 4: Classifier algorithm comparison for Any, IRC and Neris Botnet with or without IP addresses or Ports

Classifier Accuracy	IRC *		Neris*		Any*	
	With IP	Without IP*	With IP	Without IP	With IP	Without IP
LR	0.966 ± 3E-3 (11s)	0.966 ± 2E-3 (3s)	0.997 ± 1E-3 (9s)	0.996 ± 1E-3 (3s)	0.893 ± 3E-3 (9s)	0.163 ± 2E-3 (3s)
LDA	0.963 ± 2E-3 (6s)	0.962 ± 2E-3 (6s)	0.994 ± 1E-3 (6s)	0.994 ± 1E-3 (6s)	0.946 ± 1E-3 (7s)	0.939 ± 2E-3 (5s)
KNN	0.983 ± 1E-3 (53s)	0.977 ± 2E-3 (50s)	0.997 ± 1E-3 (50s)	0.997 ± 1E-3 (50s)	0.973 ± 2E-3 (50s)	0.963 ± 3E-3 (50s)
CART	**0.992 ± 1E-3 (4s)**	0.979 ± 1E-3 (7s)	**0.999 ± 1E-3 (7s)**	0.996 ± 1E-3 (6s)	**0.989 ± 1E-3 (5s)**	0.974 ± 2E-3 (7s)
RFC	0.992 ± 1E-3 (41s)	**0.982 ± 1E-3 (36s)**	0.999 ± 1E-3 (40s)	**0.999 ± 1E-3 (32s)**	0.986 ± 1E-3 (51)	**0.978 ± 2E-3 (48s)**
SVM	0.966 ± 2E-3 (68s)	0.966 ± 2E-3 (72s)	0.996 ± 1E-3 (9s)	0.996 ± 1E-3 (9s)	0.837 ± 2E-3 (646s)	0.837 ± 2E-3 (389s)
Bayes	0.086 ± 3E-3 (2s)	0.041 ± 2E-3 (2s)	0.986 ± 2E-3 (7s)	0.951 ± 2E-3 (2s)	0.163 ± 1E-3 (2s)	0.163 ± 2E-3 (2s)

*Average ± standard deviation (runtime)

Table 5: Metrics on Predicted Models for Test Dataset

RFC Metrics for Test Dataset With IPs and Ports

Botnet Traffic	Accu-racy	False			True		
		Prec	Recall	F1-S	Prec	Recall	F1-S
ANY	0.97	0.95	0.98	0.97	0.99	0.97	0.98
IRC	0.97	0.97	0.99	0.98	0.97	0.91	0.94
Neris	1.00	1.00	1.00	1.00	1.00	1.00	1.00
RBOT	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Menti	1.00	1.00	1.00	1.00	0.97	0.89	0.93
Sogou	1.00	1.00	1.00	1.00	1.00	0.38	0.56
Murlo	1.00	1.00	1.00	1.00	0.99	0.97	0.98
Virtu	1.00	1.00	1.00	1.00	0.97	0.95	0.96
BH1	1.00	1.00	1.00	1.00	0.00	0.00	0.00
BH2	1.00	1.00	1.00	1.00	0.92	0.92	0.95
BH3	1.00	1.00	1.00	1.00	0.95	0.68	0.79
TBOT	1.00	1.00	1.00	1.00	1.00	0.90	0.94
Weasel	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Zeus	1.00	1.00	1.00	1.00	0.97	0.68	0.80
OSX TJ	1.00	1.00	1.00	1.00	0.00	0.00	0.00
ZeroAccess	1.00	1.00	1.00	1.00	1.00	0.96	0.98
SmokeBot	1.00	1.00	1.00	1.00	1.00	0.56	0.72

RFC Metrics for Test Dataset Without IPs or Ports

Botnet Traffic	Accu-racy	False			True		
		Prec	Recall	F1-S	Prec	Recall	F1-S
ANY	0.92	0.88	0.93	0.90	0.95	0.92	0.93
IRC	0.93	0.94	0.97	0.95	0.88	0.79	0.83
Neris	0.98	0.99	0.99	0.99	0.92	0.85	0.88
RBOT	0.99	1.00	1.00	1.00	0.93	0.93	0.93
Menti	1.00	1.00	1.00	1.00	0.92	0.49	0.64
Sogou	1.00	1.00	1.00	1.00	0.50	0.08	0.13
Murlo	1.00	1.00	1.00	1.00	0.84	0.94	0.89
Virtu	1.00	0.99	1.00	1.00	0.96	0.93	0.94
BH1	1.00	1.00	1.00	1.00	0.00	0.00	0.00
BH2	1.00	1.00	1.00	1.00	0.98	0.92	0.95
BH3	1.00	1.00	1.00	1.00	0.98	0.66	0.79
TBOT	1.00	1.00	1.00	1.00	0.96	0.73	0.83
Weasel	1.00	1.00	1.00	1.00	1.00	0.99	0.99
Zeus	1.00	1.00	1.00	1.00	0.54	0.35	0.42
OSX TJ	1.00	1.00	1.00	1.00	0.00	0.00	0.00
ZeroAccess	1.00	1.00	1.00	1.00	0.93	0.70	0.80
SmokeBot	1.00	1.00	1.00	1.00	0.33	0.16	0.28

Table 6: TOP-10 Feature Importance for ANY Botnet With IP and Ports

Feat. Imp ANY Botnet with IPs and Ports				Feature Importance ANY Botnet without IPs or Ports			
Packet_Attribute	Score	Packet_Attribute	Score	Packet_Attribute	Score	Packet_Attribute	Score
Src IP	0.187	Flow Duration	0.028	Init Bwd Win Byts	0.090	Flow Duration	0.030
Dst IP	0.115	Flow IAT Max	0.025	Flow IAT Max	0.043	Flow IAT Mean	0.028
Init Bwd Win Byts	0.043	Subflow Fwd Byts	0.021	Subflow Fwd Byts	0.037	Bwd Header Len	0.028
Src Port	0.043	Bwd Pkt Len Min	0.021	Flow Byts/s	0.037	TotLen Fwd Pkts	0.027

SYN Flag Cnt	0.030	Dst Port	0.021	Flow IAT Min	0.031	Bwd Pkts/s	0.026

Analysis per Botnet. As we can see on Table 5, on major cases we gain accuracy, precision, recall and F1-Score on each Botnet versus all Botnet. On some particular cases, such as IRC, BH3 and Zeus we lose Recall on *True* class. We suspect that the average metrics of all Botnets overfits the low metrics of some individual Botnet. Analysis with and without IP addresses and Ports. We can note that the IP address and Port information holds an higher accuracy, but lacks on precision and recall (and consequently F1-Score). The True class of each Botnet on both scenarios has depreciated metrics if compared to the respective false class.

On BH1 and OSX Botnet traffics with and without IP addresses or L4 Ports. As described on Table 5, on this test we obtained 0.00 F1-score on True Class, indicating no True prediction on this classification. This shows that this Botnet traffic might not be well represented on the selected dataset. Similar scores were obtained with Sogou Botnet on both scenarios and with Zeus on scenario without IP addresses or Ports.

4.2. Feature Importance Selection

On the table 6, the comparison of the TOP-10 feature selection for ANY Botnet With and Without IP addresses. Note how overfitting the IP address and Ports can be. Once removed, other fields gained priority (Such as Subflow Fwd Byts), other kept its priority. Important to say that this is an average feature of all Botnets.

Analysing the feature importance on Specific Botnet, the features might have different priorities. Just for illustrating, on Table 7 we can see three different Botnet traffics without IP or Port Addresses. Note that some of these packet attributes can be seen on particular Botnets (Such as Forwarded and Backwarded Packets and Inter Arrival Time) as can also be seen on the Any Botnet Feature Importance.

The added fields “Dst Port Category” and “Src Port Category” also were prioritised on the Murto and SmokeBot Botnet traffics. Interesting that none of these fields are on the TOP-10 of the BlackHole1 Botnet traffic.

Table 7: TOP-10 Feature Importance for Each Botnet Without IP or Ports

Feat. Imp. Murto w/o IPs		Feat. Imp BH1 w/o IPs		Feat. Imp.SmokeBot w/o IPs	
Packet_Attribute	Score	Packet_Attribute	Score	Packet_Attribute	Score
Init Bwd Win Byts	0.122	Flow Duration	0.074	Init Bwd Win Byts	0.146
Bwd Header Len	0.109	Flow IAT Min	0.072	Flow IAT Min	0.059
Fwd Pkts/s	0.061	Init Bwd Win Byts	0.063	Flow IAT Mean	0.055
Flow IAT Max	0.060	Flow Pkts/s	0.058	Fwd Pkts/s	0.05
Flow IAT Mean	0.053	Flow IAT Mean	0.048	Bwd Pkts/s	0.045
Bwd Pkts/s	0.048	Bwd Pkts/s	0.042	Flow Pkts/s	0.042
Dst Port Category	0.047	Bwd Header Len	0.041	Flow IAT Max	0.038
Flow Pkts/s	0.046	TotLen Fwd Pkts	0.036	Fwd IAT Max	0.032
ACK Flag Cnt	0.035	Tot Bwd Pkts	0.032	Flow Duration	0.032
Flow IAT Min	0.035	Fwd IAT Mean	0.027	Fwd Pkt Len Std	0.027

5. Conclusion

Random Forest (RF) and Decision Tree (CART) were the best algorithms in accuracy, according to the tests executed for Classifier algorithm selection. If performance might be a problem, CART would be a better solution than RF. We had reached with CART similar accuracy with 10-20% of RF run time. The decision for RF Classifier was due to previous works that shows better accuracy on overall cyberattack types.

Including the IP address and Port information on the ML analysis showed that we gained accuracy but lost some F1-Score. Note that the source and destination IP addresses and Ports are very mutable data, that probably might not be useful as a feature. Neverless, this information is very useful as intermediate, essential for identifying the IP Flows.

The analysis of the L4 Port privileges on the host systems showed that it is a feature relevant for certain Botnet traffics, such as Murto and SmokeBot, but is not a relevant feature for all Botnet traffics. Besides that, the differences on the metrics between each Botnet classification shown that some Botnet traffic need more representative data. Also,

the feature relevance differences confirms the literature. It shows that Botnet traffic is not a single uniform protocol, but a collection of very different ways of communications between the botmaster and the infected host.

As next steps, develop the ML for multiclass classification and the algorithm for neural networks. Compare it with the results obtained from ML. Also, as some of the Botnet traffic is not well represented, study its behaviour would help to identify some additional features for its identification.

Acknowledgements

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020.

References

[1] Aleesa, A. M., Zaidan, B. B., Zaidan, A. A., and Sahar, N. M. (2019). Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. Number October. Springer London.

[2] Anagnostopoulos, M., Kambourakis, G., and Gritzalis, S. (2016). New facets of mobile botnet: architecture and evaluation. International Journal of Information Security, 15(5):455 473.

[3] Beigi, E. B., Jazi, H. H., Stakhanova, N., and Ghorbani, A. A. (2014). Towards e ective feature selection in machine learning-based botnet detection approaches. 2014 IEEE Conference on Communications and Network Security, CNS 2014, pages 247 255.

[4] Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martinez-del Rincon, J., and Siracusa, D. (2020). LUCID: A Practical, Lightweight Deep Learning Solution for DDoS Attack Detection. IEEE Transactions on Network and Service Management, pages 1 1.

[5] Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., and Ghorbani, A. A. (2016). Characterisation of encrypted and VPN traffic using time-related features. ICISSP 2016 - Proceedings of the 2nd International Conference on Information Systems Security and Privacy, (Icissp):407 414.

[6] Ferrag, M. A., Maglaras, L., Janicke, H., and Smith, R. (2019). Deep Learning Techniques for Cyber Security Intrusion Detection : A Detailed Analysis. pages 126 136.

[7] Haddadi, F. and Zincir-Heywood, A. N. (2017). Botnet behaviour analysis: How would a data analytics-based system with minimum a priori information perform? International Journal of Network Management, 27(4):1 19.

[8] Hanzlik, L., Kuty“owski, M., and Yung, M. (2015). Information Security Practice and Experience. Lecture Notes in Computer Science, 9065(June):421 436.

[9] IANA (2008). IANA Allocation Guidelines for TCP and UDP Port Numbers.

[10] Islam, S. R., Eberle, W., Ghafoor, S. K., Siraj, A., and Rogers, M. (2019). Domain Knowledge Aided Explainable Arti cial Intelligence for Intrusion Detection and Response. CEUR Workshop Proceedings, 2600.

[11] Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1).

[12] Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., and Ghorbani, A. A. (2017). Characterisation of Tor Traffic using Time based Features. In ICISSP, pages 253 262.

[13] Peng, T., Leckie, C., and Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the DoS and DDoS problems. ACM Computing Surveys, 39(1).

[14] Pfahringer, B. (2000). Winning the KDD99 classification cup: bagged boosting. ACM SIGKDD Explorations Newsletter, 1(2):65 66.

[15] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., and Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers and Security, 86:147 167.

[16] Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterisation. ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy, 2018-Janua(Cic):108 116.

[17] Shiravi, A., Shiravi, H., Tavallaee, M., and Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers and Security, 31(3):357 374.

[18] Srusti D. Mehta and Deepak Upadhyay (2020). A Review on Classification of Tor-Nontor Traffic and Forensic Analysis of Tor Browser. International Journal of Engineering Research and, V9(04):776 778.

[19] T., S. and S., M. (2011). Advanced Methods for Botnet Intrusion Detection Systems. Intrusion Detection Systems.

[20] Tanenbaum, A. S. and Wetherall, D. (2011). Computer Networks. Pearson Prentice Hall.

[21] Tao, W., Zhang, W., Hu, C., and Hu, C. (2020). A Network Intrusion Detection Model Based on Convolutional Neural Network. Advances in Intelligent Systems and Computing, 895(4):771 783.

[22] Ullah, I. and Mahmoud, Q. H. (2020). A two-level ow-based anomalous activity detection system for IoT networks. Electronics (Switzerland), 9(3).

[23] Venturi, A., Apruzzese, G., Andreolini, M., Colajanni, M., and Marchetti, M. (2021). DReLAB - Deep REinforcement Learning Adversarial Botnet: A benchmark dataset for adversarial attacks against botnet Intrusion Detection Systems. Data in Brief, 34:106631.

[24] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., AlNemrat, A., and Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access, 7:41525 41550.

[25] Vormayr, G., Zseby, T., and Fabini, J. (2017). Botnet Communication Patterns. IEEE Communications Surveys and Tutorials, 19(4):2768 2796.
[26] Xing, Y., Shu, H., Zhao, H., Li, D., and Guo, L. (2021). Survey on Botnet Detection Techniques: Classification, Methods, and Evaluation. Mathematical Problems in Engineering, 2021.
[27] Zamani, M. and Movahedi, M. (2013). Machine Learning Techniques for Intrusion Detection. pages 1 11.
[28] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., and Garant, D. (2013). Botnet detection based on traffic behavior analysis and ow intervals. Computers and Security, 39(PARTA):2 16.
[29] Zhou, Q. and Pezaros, D. (2019). Evaluation of Machine Learning Classifiers for Zero-Day Intrusion Detection An Analysis on CIC-AWS-2018 dataset.
[30] (n.d.). TensorFlow. https://www.tensorflow.org/