Botnet detection based on network flow analysis using inverse statistics

Daniele A. G. Lopes, Marcelo A. Marotta, Marcelo Ladeira, João J. C. Gondim
Department of Computer Science
University of Brasilia (UnB)
Brasilia, Brazil
daniele.lopes@aluno.unb.br, {marcelo.marotta, mladeira, gondim}@unb.br

Abstract — **A botnet is a network of infected computers, which are remotely controlled by a cybercriminal, called botmaster, which aims to carry out massive cyberattacks, such as DDoS, SPAM, and information theft. Traditional botnet detection methods, usually signature-based, are unable to detect unknown botnets. The behavior-based analysis is promising for detecting current botnet trends, which are constantly evolving. This article proposes an exploration analysis of botnet detection mechanisms based on the network flow behavior. The main technique used to detect botnets was recently developed and is called Energy-based Flow Classifier (EFC). This technique uses inverse statistics to detect anomalies. Two heterogeneous datasets, CTU-13 and ISOT HTTP were used to evaluate the efficiency of the generated model and the results were compared with several traditional classifiers, of one and two classes. The results obtained show that EFC obtained more stable results, regardless of the domain, unlike the other tested algorithms.**

Keywords – botnet; network flow; anomaly detection; inverse statistics.

I. INTRODUCTION

A botnet is a network formed by numerous devices infected by some malware, which are called bots or zombies and which are controlled by an attacker, called botmaster [1]. The purpose of a botnet is to carry out malicious activities based on instructions provided by the botmaster. The main component of a botnet is the Command and Control (C&C) server, as it is the means by which the botmaster controls and sends instructions to the bots, initiating various types of cyber attacks such as distributed denial of service (DDoS), spam, phishing and information theft [2]. The C&C channel structure can be centralized, in which a central C&C server is responsible for sending commands to the bots, or decentralized (P2P), where infected devices act as bots and as C&C servers at the same time [3].

The destructive potential of botnets has increased exponentially with the advancement of the Internet of Things (IoT) technology and as the number of connected users and devices has increased [4]. In 2016 botnet Mirai was responsible for one of the highest distributed denial of service attacks ever recorded to date, estimated at 1.2 Tbps (terabits per second). This attack has shut down sites like Twitter, Netflix, CNN, and others across Europe and the United States [5] [6]. With the availability of Mirai's source code on the Internet, many variant projects have emerged. In 2019, for example, the number of variants of botnet Mirai had a growth of 57% compared to 2018, surpassing 225,000 occurrences [7].

Since traditional botnets detection methods are signature based, they become efficient to detect known types of botnets. However, new types of botnets or variants of botnets emerge daily and their detection is a major challenge for traditional methods focused on signatures of known attacks [1]. In addition, botnets are constantly evolving, changing their architecture and protocols used, to avoid detection by security systems. Additionally, botnets increasingly use encryption and obfuscation techniques, making detection even more difficult [2]. As botnets have progressed and become more complex, various botnet detection strategies have been proposed, mainly using machine learning methods for behavior analysis and anomaly detection [8].

In the context of attack detection by botnets, most methods differ in the type of analysis performed, being *(i)* deep packet analysis or *(ii)* flow analysis. In the first one, the packets are individually analyzed considering their header and the data being transported (payload). In the second, a set of packets are grouped according to common characteristics present in their headers, called flows, which are evaluated according to these characteristics and statistical metrics, such as the number of bytes and average duration time. Flow analysis has some advantages over deep packet analysis, mainly because it consumes less computational resources since it only processes the packet headers. Also, the fact that it allows the detection of botnets that use techniques of encryption or obfuscation, as it does not require access to the payload of the package, which may be encrypted [2]. Therefore, flow analysis will be the focus of our study.

Many approaches have been proposed in recent years for detecting botnets based on network flow using machine learning techniques [2] [1] [9]. However, some techniques are made especially for specific protocols and structures, being unable to detect botnets that use different protocols or structures [10] [11] [12]. Furthermore, most works use conventional machine learning algorithms, such as Support Vector Machine (SVM) and Naive Bayes (NB) [13]. These algorithms are based on learning from a training set that contains samples of the two classes (benign flows and malicious flows). Therefore, only botnets found during training or with very similar behavior will be detected, limiting the objective of detecting unknown botnets. Consequently, most existing approaches do not adapt well to different domains, i.e., performance is reduced when trained on a specific dataset and evaluated on another related dataset [14] [15]. Finally, it is not easy to get samples of malicious flows that represent recent malware to compose the training set and make models based on two classes more efficient [10].

2022 17th Iberian Conference on Information Systems and Technologies (CISTI)
22 – 25 June 2022, Madrid, Spain
ISBN: 978-989-33-3436-2

In a recent study, Pontes et al. [16] developed a new classifier, called Energy-Based Flow Classifier (EFC), which was inspired by the inverse Potts model of quantum mechanics and adapted for the classification of network flows. EFC is an algorithm that performs one-class classification using only benign data to carry out the training and does not need to know the behavior of malicious traffic to perform anomaly detection, thus circumventing the problem of obtaining labeled malicious samples. Furthermore, EFC is an intrinsically adaptable classifier to different domains, since the model's inference is based only on benign samples [16]. Due to this characteristic, EFC seems to be a promising classifier for detecting new types of botnets or even variants of known botnets, but not yet explored in Pontes's et al. work.

Therefore, we propose to evaluate the Energy-Based Flow Classifier (EFC) algorithm for detecting botnets by analysis of network flows, proposing an approach capable of detecting new types of botnets, regardless of the structure or protocols used. To evaluate the model's efficiency, we will use two heterogeneous datasets (CTU-13 and ISOT HTTP). Also, we will compare EFC's performance with traditional classifiers of one and two classes. Our results show that EFC proved to be more robust and less sensitive to changes in the data distribution than other algorithms. Our main contributions are:

- An exploratory botnets detection analysis using the EFC algorithm;
- A comparison of EFC performance with classical classifiers of one and two classes using two different datasets;
- An analysis of the adaptability of the different classifiers when tested in a domain other than the one where the training took place.

The remainder of this paper is organized as follows. Section 2 presents the works related to the research topic. Section 3 presents the methodology and details for understanding the EFC, also describes the datasets used. Section 4 presents the results obtained. Finally, Section 5 concludes the work and directs future work.

II. Related Work

Many approaches have been proposed in recent years for detecting botnets based on the analysis of network flows using machine learning. Most of these approaches were developed considering a specific protocol type, as in [10], [11], and [12], which focused on the detection of P2P botnets. In [10], a framework for the detection of botnets has been proposed that has two phases. The first implements the pre-processing of network traffic, extracting a wide set of attributes for each flow. The second phase implements supervised models to classify flows into non-P2P traffic, malicious P2P traffic, and normal P2P traffic. The following algorithms were tested: SVM (Support Vector Machine), ANN (Artificial Neural Network), Nearest Neighbor classifier, Gauss-based classifier, and the Naive Bayes classifier. The authors concluded that all five techniques provide a detection rate greater than 89%, but ANN and SVM require more time to be trained and to perform the classification. According to the authors themselves, the proposed approach is not capable of adapting to changes in network traffic, nor of detecting new types of botnets [10]. Differently, our approach aims to detect botnets regardless of the protocol used, in addition to seeking the detection of unknown botnets.

Others researches addressed the detection of botnets based on the structure of the C&C channel, as was done in [21], where the authors proposed a model of botnet detection called BotCap, using the SVM and J48 algorithms to train the model. The dataset was generated by the authors, with a total of six families of botnets, all with a centralized architecture (HTTP and IRC), so the detection of P2P botnets was not considered. Our work uses two publicly available datasets (CTU-13 and ISOT HTTP), having centralized and decentralized architecture botnets and the protocols HTTP, IRC, and P2P.

Several other studies were developed to detect botnets independent of the protocol and architecture of C&C [22] [2]. In [22], a system for detecting C&C servers (defined as an IP and port pair) was proposed, independent of the protocol used by botnets. The attributes used for detection were extracted from Netflow data and were categorized into three groups: based on flow size, based on client access patterns, and based on temporal attributes. The following models were evaluated: Random Forest, J48 decision tree, and SVM. To reduce the false-positive rate, a series of reputation lists (blacklists) were incorporated into the detection procedure. The approach was tested on two real-world networks, with an identification rate of true positive of 65% and a false positive rate of 1%.

Ibrahim et. al. proposed a multi-layered framework for detecting Command and Control servers of botnets. The approach consists of two main modules. The first is the filtering module that has the objective of filtering and reducing the network traffic for the second module, using the k-means clustering algorithm. The objective of the second module is to detect the C&C server, using classification algorithms. Three classifiers were evaluated: KNN, SVM, and Multilayer Perceptron, being that KNN presented the best result with 91.51% of F1-score and a false negative rate of 1.5%.

The algorithms used in the cited work perform the classification of traffic based on learning from samples of the two classes (benign and malicious). Therefore, it is necessary to know the malicious behavior to perform the detection, limiting the detection of unknown botnets. Differently, in our work we will use a unary flow classification algorithm, called Energy based Flow Classifier (EFC). This algorithm was proposed by Pontes et al. [16] with the purpose of overcoming some limitations of machine learning algorithms, such as the need for large amounts of categorized examples, and especially the fact that most of these algorithms is not easily generalizable to other datasets, i.e., performance is reduced when trained on a specific dataset and evaluated on another dataset [16]. Due to the results obtained by Pontes et al. [16] in the detection of network anomalies, we will use the EFC specifically for the detection of botnets, aiming to detect new types of botnets or even variants of known botnets. Finally, we will implement most of the algorithms used in the related work for comparison with EFC.

ISBN: 978-989-33-3436-2

III. METHODOLOGY

In this section, we present the main concepts about the EFC algorithm and then describe the two datasets used in our experiments. Finally, we present the details of pre-processing the datasets.

A. Energy-based Flow Classifier - EFC

EFC is a classifier that uses inverse statistical techniques to, during the training stage of the model, infer a probability distribution for the class of flows to be detected, based only on samples of benign flows. In the model testing step, the distribution defined in the previous step is used to classify new flows by calculating and comparing a measure called flow energy, which measures how unlikely is the occurrence of a flow in the calculated distribution [16].

If the energy of the flow is high, i.e., below a certain threshold, it means that it does not resemble the flows that generated the distribution. Likewise, if the energy is low, this flow is more likely to exist in the distribution. The EFC threshold is set based on the energies of the training samples and can be set dynamically or statically. In our case study, a statistical threshold defined by the 95th percentile of the energies of the training samples was used. Thus, if a flow has an energy value below the 95th percentile of the benign samples, that is, below the threshold, it is considered normal. Otherwise, it is classified as malicious traffic. The theoretical details of the model inference are presented in Pontes et al. [16].

B. Datasets

The two datasets used in this research will be described below and were selected because they are popularly used in the literature due to their relevance and realism [11] [2].

1) CTU-13: CTU-13 is a botnet traffic dataset that was captured at CTU University, Czech Republic, in 2011 and stored in PCAP files [17]. The CTU-13 dataset contains 13 traffic capture files, which are called scenarios and are labeled as Normal, Attack, or Background. These files contain different types of botnets, including centralized (IRC and HTTP) and decentralized (P2P) structures and various protocols. Thus, this dataset served our purpose of designing a botnets detection model that was structure and protocol independent.

We used in our experiments the files corresponding to scenarios 1, 3, 5, 6, 7, 8, and 12. Consequently, we tested seven types of botnets: Neris, Rbot, Virut, Menti, Sogou, Murlo, and Nsis.ay, where the combination of these botnets consisted of both centralized and decentralized structures. Due to privacy concerns, the PCAP files made available contain only malicious traffic, whereas the complete capture containing all background, normal, and botnet data are not openly available. Therefore, we use part of the ISCXIDS-2012 project dataset[1] to obtain only normal traffic data to complement the dataset CTU-13 [18]. In this case, we used the PCAP file referring to the capture of 6 December 2010 (Saturday), which has 4.22 GB of normal traffic.

2) ISOT HTTP Botnet: The ISOT HTTP dataset[2] was made available by the University of Victoria, Canada and is composed of two sets of different data. The first consists of malicious traffic generated by different botnets, while the second consists of benign traffic generated by various software applications such as antivirus, online chat, and instant messaging applications (i.e Skype, Facebook, Messenger) [19]. The capture of traffic from the two environments (normal and botnet) was carried out from 14 June to 21 June 2017.

Malicious traffic was collected from a virtual environment, in which different kits of exploits for HTTP botnets were implemented, totaling 9 (nine) command and control (C&C) servers, one for each type of botnet, which generated 5 (five) PCAP files. Benign traffic was also captured from a virtual environment simulating traffic from several applications installed on virtual machines configured with Windows 7 operating system, resulting in 3 (three) PCAP files.

We used in our experiments the 3 files containing benign traffic and for malicious traffic, we used only the init4.pcap file, since this single file contains traffic from all types of botnets present in the ISOT HTTP dataset. The following types of botnets are present in the used file: zyklon, blue, liphyra, gaudox, blackout, citadel, be.botnet, and zeus. All these botnets have a centralized architecture and use the HTTP protocol.

C. Feature Extraction

To extract the network flows from the traffic captures referring to the two datasets described above, we used the CICFlowMeter[3] tool, which is a network traffic flow generator and analyzer provided by the Canadian Institute for Cybersecurity [20]. The generated result is a CSV file containing 84 features with traffic statistics, e.g., total, average and minimum packets sent and received. All the generated features were used for the initial experiments, except of the Flow ID, Source IP, Destination IP, and Timestamp features, as they are considered very specific for each flow. Also, for CICFlowMeter, each flow is defined by the first packet that determines the forward (source to destination) and backward (destination to source) directions. Further, a flow is finished for TCP flows upon connection teardown (by FIN packet) while UDP flows are terminated exclusively by flow timeout that is determined arbitrarily for both, usually 600 seconds for TCP and UDP.

After extracting the features, we labeled the resulting files from each dataset using the python programming language and pandas library. Table I shows the final composition of the two datasets, including the amount of extracted benign and malicious flows and also the amount per botnet family. Due to space limitation, we did not present the 84 features extracted in the table, but the reader can refer to [3] for a list these features. Further, as EFC works with discretized data to carry out the classification, we discretize the data only for the EFC's implementation. For the other models used in this research, we normalized the data since discretization could harm the performance of these algorithms.

[1] https://www.unb.ca/cic/datasets/ids.html
[2] https://www.uvic.ca/ecs/ece/isot/datasets/botnet-ransomware/index.php
[3] https://www.unb.ca/cic/research/applications.htmlCICFlowMeter

ISBN: 978-989-33-3436-2

TABLE I: QUANTITATIVE OF THE DATASETS

CTU-13		ISOT HTTP	
Label	**Quantity**	**Label**	**Quantity**
Normal	215.251	Normal	76.360
Virut	83.900	Cidatel	145.087
Rbot	46.540	Gaudox	90.970
Neris	22.247	Zeus	80.642
Murlo	11.536	Be.botnet	13.755
Nsis	7.645	Bluebot	13.593
Menti	4.809	Zyklon	12.008
Sogou	72	Blackout	6.881
		Liphyra	3.782
Total Malicious	176.749	**Total Malicious**	366.718
Total Benign	215.251	**Total Benign**	76.360

The EFC's performance was compared to the performance of the most popular algorithms for detecting anomalies based on the analysis of network flows: K-Nearest Neighbors (KNN), Decision Tree (DT), Multilayer Perceptron (MLP), Naive Bayes (NB), and Support Vector Machine (SVM), in addition to the ensemble classifiers: AdaBoost (AD) and Random Forest (RF). Furthermore, as EFC is a one-class algorithm, that is, it is trained only with benign traffic, experiments were also carried out with the following One Class algorithms available in the scikit-learn[4] library: One-Class SVM (OCSVM), Isolation Forest (iForest), Local Outlier Factor (LOF), and Eliptic Envelop (Elenv). The performance of the classifiers used in this research was measured based on the mean of the AUC and F1-score values over 5 test sets (5 Stratified k-fold) and on the standard error, with a confidence interval of 95%. Finally, it should be noted that we implemented all models using the standard scikit-learn configuration.

All experiments carried out in this work were executed on a notebook with the following configuration: Intel Core I7-7700HQ 3.8 GHz processor, 32 GB of RAM, and Linux Debian 10 operating system. We performed two principal tests. The first is intra-domain testing, in which training and testing of models were performed using the same dataset. The second is cross-domain testing, in which models are trained on one dataset and evaluated on another one. The objective of the second test is to assess the ability of models to adapt to changes in the network and, consequently, the ability to detect unknown botnets.

IV. RESULTS

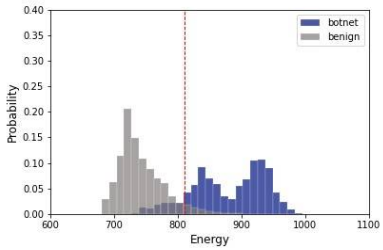
In this section, we present the results obtained using the EFC classifier to detect traffic related to botnet activity, as well as the results obtained using several classifiers of one and two classes.

A. Distribution of Calculated Energies by the EFC

EFC infers a statistical model based on benign flows samples during model training. Further, the trained model is used to calculate the energies of samples of benign and malicious flows contained in the test set to classify them.

Fig. 1 (a) illustrates the energy values calculated considering part of the benign samples from the CTU-13 dataset, and Fig. 1 (b) shows the energy values for the ISOT HTTP dataset. These values refer to intra-domain testing, i.e., training and testing on the same dataset. It can be seen that the separation between the two classes is clear, i.e., the energy of benign flows is shifted to the left compared to the distribution of the energies of malicious flows. The red vertical line represents the EFC's classification threshold, which was defined as the 95th percentile of the energy distribution obtained in the training stage.

1 (a) Train/Test CTU-13



1 (b) Train/Test ISOT HTTP

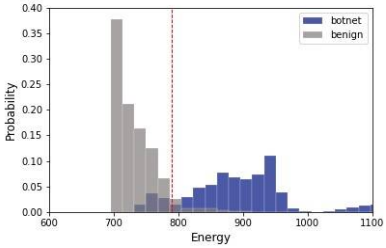


Figure 1. Histograms of Calculated Energies in the Test Phase Using the CTU-13 and ISOT HTTP Datasets.

B. EFC x Two-Class Classification Algorithms

Table II shows the average performance and standard error (with a confidence interval of 95%) for each classifier, using the ISOT HTTP dataset. In the first approach, which is the intra-domain test, all the proposed models obtained very similar results, with an F1-score above 0.98 and AUC above 0.99. The only exception was for the NB algorithm, which had the lowest performance, with an F1-score of 0.952 ± 0.000 and AUC of 0.799 ± 0.004. In the inter-domain test, where the ISOT HTTP dataset was used for training and the CTU-13 for testing, the EFC obtained a much higher result when compared to the other models, mainly concerning the F1-score metric (0.663 ± 0.001). A fact that caught our attention was that the NB, RF, and AD models obtained an F1-score equal to 0. It was observed through the confusion matrices, as shown in Fig. 2, that these models classified all malicious instances as benign, thus obtaining a true-positive rate equal to 0, which explains the value of the F1-score obtained by these models. Finally, since the ensemble classifiers use a combination of predictions from different models and generally perform better than simple classifiers, it was expected that the performance of RF and AD would be superior to the performance of the other models tested. The inter-domain test in this scenario is quite challenging since the ISOT HTTP dataset has only instances of botnets that use the HTTP protocol for communication (centralized architecture), while the CTU-13 dataset has botnets that use HTTP, IRC, and P2P protocols (centralized and decentralized architectures). Thus, the

[4] https:// scikit-learn.org/stable/

ISBN: 978-989-33-3436-2

result obtained by the EFC was quite satisfactory, given the considerable change in context.

TABLE II: AVERAGE PERFORMANCE OF CLASSIFIERS – ISOT HTTP

	Training/Test ISOT		Training ISOT/Test CTU-13	
Classifier	**F1-score**	**AUC**	**F1-score**	**AUC**
NB	0.952 ± 0.000	0.799 ± 0.004	0.000 ± 0.000	0.500 ± 0.000
KNN	**0.999 ± 0.000**	0.997 ± 0.000	0.074 ± 0.009	0.333 ± 0.009
DT	**0.999 ± 0.000**	0.996 ± 0.000	0.019 ± 0.031	0.473 ± 0.031
SVM	0.989 ± 0.000	0.998 ± 0.000	0.120 ± 0.002	0.636 ± 0.002
MLP	0.994 ± 0.001	**0.999 ± 0.000**	0.271 ± 0.240	0.601 ± 0.240
EFC	0.989 ± 0.000	0.995 ± 0.000	**0.663 ± 0.001**	0.535 ± 0.001
Ensemble				
RF	**0.999 ± 0.000**	**1.000 ± 0.000**	0.000 ± 0.000	0.539 ± 0.000
AD	0.998 ± 0.001	**1.000 ± 0.000**	0.000 ± 0.000	**0.756 ± 0.000**

The results using the CTU-13 dataset can be seen in Table III. In the first approach, which is the intra-domain test, the KNN, DT, MLP, and RF models obtained an F1-score and AUC above 0.99, while the EFC obtained an F1-score of 0.877 ± 0.000 and AUC of 0.961 ±0.000. Over again NB achieved the lowest performance, with an F1-score of 0.677 ± 0.001 and AUC of 0.864 ± 0.013.

TABLE III: AVERAGE PERFORMANCE OF CLASSIFIERS – CTU-13

	Training/Test CTU-13		Training CTU-13/Test ISOT	
Classifier	**F1-score**	**AUC**	**F1-score**	**AUC**
NB	0.677 ± 0.001	0.864 ± 0.013	0.109 ± 0.214	0.675 ± 0.214
KNN	0.997 ± 0.000	0.999 ± 0.000	0.005 ± 0.007	0.457 ± 0.007
DT	**0.999 ± 0.000**	**0.999 ± 0.000**	0.544 ± 0.004	0.275 ± 0.004
SVM	0.912 ± 0.003	0.962 ± 0.001	0.481 ± 0.010	0.665 ± 0.003
MLP	0.994 ± 0.000	1.000 ± 0.000	0.325 ± 0.240	0.711 ± 0.240
EFC	0.877 ± 0.000	0.961 ± 0.000	0.758 ± 0.076	**0.729 ± 0.076**
Ensemble				
RF	**0.999 ± 0.000**	**1.000 ± 0.000**	**0.794 ± 0.022**	0.702 ± 0.022
AD	0.980 ± 0.002	0.998 ± 0.000	0.262 ± 0.172	0.573 ± 0.172

In the inter-domain test, where the CTU-13 dataset was used for training and the ISOT-HTTP for testing, the Random Forest obtained the highest F1-score value (0.794 ± 0.022), while the EFC obtained the best AUC (0.729 ± 0.076). The performance of the other classifiers was much lower. Inter-domain testing in this scenario is less challenging when compared to the previous experiment. Here, training is performed on a dataset containing botnets with various communication protocols (HTTP, IRC, and P2P) and testing on a dataset containing only HTTP botnets. Thus, all models obtained a performance superior to that obtained in the inter-domain previous experiment, and it is observed that none of the models obtained F1-score equal to 0.

C. EFC X One-Class Classification Algorithms

Table IV shows the average performance and standard error (with a confidence interval of 95%) of each classifier, using the ISOT-HTTP dataset. In the first approach, which is the intra-domain test, the EFC performed well above the other classifiers, both concerning the F1-score (0.989 ± 0.000) and the AUC (0.995 ± 0.000). The lowest performance was obtained by the Elenv classifier with an F1-score of 0.035 ± 0.034 and AUC of 0.781 ± 0.005. In the inter-domain test, where the ISOT HTTP dataset was used for training and the CTU-13 for testing, the EFC kept the best performance concerning the F1-score (0.663 ± 0.001), followed by the OCSVM classifiers (0.627 ± 0.005) and LOF (0.621 ± 0.000). Regarding the AUC, the LOF obtained the best result (0.770 ± 0.000), followed by the OCSVM (0.726 ± 0.005) and by the EFC (0.535 ± 0.001).

TABLE IV: AVERAGE PERFORMANCE OF ONE-CLASS CLASSIFIERS – ISOT HTTP

	Training/Test ISOT		Training ISOT/Test CTU-13	
Classifier	**F1-score**	**AUC**	**F1-score**	**AUC**
EFC	**0.989 ± 0.000**	**0.995 ± 0.000**	**0.663 ± 0.001**	0.535 ± 0.001
OCSVM	0.731 ± 0.001	0.540 ± 0.002	0.627 ± 0.005	0.726 ± 0.005
iForest	0.670 ± 0.040	0.768 ± 0.009	0.443 ± 0.043	0.342 ± 0.043
Elenv	0.035 ± 0.034	0.781 ± 0.005	0.234 ± 0.247	0.466 ± 0.247
LOF	0.630 ± 0.011	0.721 ± 0.029	0.621 ± 0.000	**0.770 ± 0.000**

The results using the CTU-13 dataset can be seen in Table V. In the first approach, that is the intra-domain test, the LOF model had the best performance, both in F1-score (0.923 ± 0.002), and in AUC (0.983 ± 0.000). EFC got the second-best results with an F1-score of 0.879 ± 0.003 and AUC of 0.962 ± 0.001. The iForest classifier had the worst performance, with an F1-score of 0.084 ± 0.002 and AUC of 0.595 ± 0.025.

TABLE V: AVERAGE PERFORMANCE OF ONE-CLASS CLASSIFIERS – CTU-13

	Training/Test CTU-13		Training CTU-13/Test ISOT	
Classifier	**F1-score**	**AUC**	**F1-score**	**AUC**
EFC	0.879 ± 0.003	0.962 ± 0.001	0.705 ± 0.002	**0.736 ± 0.003**
OCSVM	0.762 ± 0.001	0.751 ± 0.003	**0.906 ± 0.000**	0.338 ± 0.001
iForest	0.084 ± 0.002	0.595 ± 0.025	0.732 ± 0.198	0.628 ± 0.002
Elenv	0.257 ± 0.143	0.705 ± 0.043	0.411 ± 0.314	0.412 ± 0.143
LOF	**0.923 ± 0.002**	**0.983 ± 0.000**	0.893 ± 0.001	0.435 ± 0.002

In the inter-domain test, where the CTU-13 dataset was used for training and the ISOT-HTTP for testing, the OCSVM obtained the best F1-score (0.906 ± 0.000), but the AUC was the lowest among all others classifiers (0.338 ± 0.001). On the other hand, the EFC obtained the best AUC (0.736 ± 0.003).

V. CONCLUSION AND FUTURE WORK

In this work, a new approach to detecting botnets was proposed through the use of the Energy-based Flow Classifier (EFC), which has as principal characteristic its adaptability to different domains [16]. In addition, we compared the performance of the EFC with various classifiers of one and two classes. The partial results obtained demonstrated that the models based on two classes suffered strong variations in the inter-domain tests, mainly in the more challenging scenario, in which the training set had only centralized botnets (HTTP) and the test set had botnets with centralized architectures (HTTP and IRC) and decentralized (P2P). In this scenario, the EFC was far superior to the other models, obtaining an F1-score of 0.663 ± 0.0010 and an AUC of 0.535 ± 0.001.

ISBN: 978-989-33-3436-2

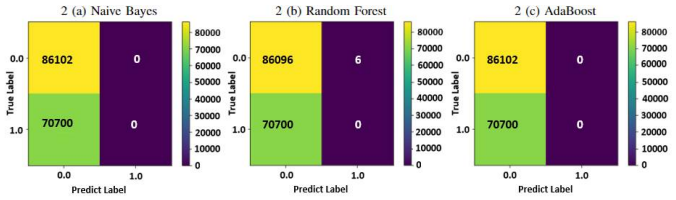


Figure 2. Confusion Matrices of NB, RF, and AD Classifiers in Inter-Domain Classification Experiments.

Regarding the models based on one class, in the intra-domain test, the EFC presented better results than the other algorithms considering the ISOT HTTP dataset, obtaining an F1-score of 0.989 and an AUC of 0.995. In tests with the CTU-13 dataset, the EFC obtained an F1-score of 0.879 and an AUC of 0.962, being surpassed only by the LOF algorithm. In the inter-domain test, the EFC was superior, either in relation to the F1-score (0.663 ± 0.001) in one of the experiments or in relation to the AUC (0.736 ± 0.003) in another experiment. Thus, in the general context, the EFC was the model that proved to be less sensitive to changes in data distribution, presenting more robust results and proving to be a promising classifier for detecting new types of botnets. As a future work, we intend to carry out an analysis and selection of the attributes that best characterize the behavior of the botnet activities, through the exploration of the EFC's ability to interpret the importance of pairs of attributes, thus aiming to obtain a better performance with the proposed model. In addition, we intend to evaluate the performance of the EFC in detecting newer botnets, using other publicly available datasets.

ACKNOWLEDGMENT

Thanks to Brazilian Army for enabling this research through the Mobile Coordination Operations Center (CCOp Mv) Project. The work developed by Prof. Marotta and Prof. Gondim is part of the project "Programmable Future Internet for Secure Software Architectures" under process nº 2020/05152-7, from Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

REFERENCES

[1] D. Zhao et al., "Botnet detection based on traffic behavior analysis and flow intervals," Comput. Secur., vol. 39, no. PARTA, pp. 2–16, 2013, doi: 10.1016/j.cose.2013.04.007.

[2] W. N. H. Ibrahim et al., "Multilayer Framework for Botnet Detection Using Machine Learning Algorithms," IEEE Access, vol. 9, pp. 48753–48768, 2021, doi: 10.1109/ACCESS.2021.3060778.

[3] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles, "Botnets: A survey," Comput. Networks, vol. 57, no. 2, pp. 378–403, 2013, doi: 10.1016/j.comnet.2012.07.021.

[4] Council to Secure the Digital Economy, "International Botnet and Iot Security Guide 2020," 2019. [On-line]. Available: https://securingdigitaleconomy.org/wpcontent/uploads/2019/11/CSDE Botnet-Report 2020 FINAL.pdf.

[5] P. Wainwright and H. Kettani, "An analysis of botnet models," ACM Int. Conf. Proceeding Ser., pp. 116–121, 2019, doi: 10.1145/3314545.3314562.

[6] M. Antonakakis et al., "Understanding the Mirai Botnet This paper is included in the Proceedings of the Understanding the Mirai Botnet," USENIX Secur., pp. 1093–1110, 2017.

[7] European Union Agency for Network and Information Security, "Botnet - ENISA Threat Landscape 2019/20," 2020.

[8] Gernot, T. Zseby, and J. Fabini, "Botnet Communication Patterns," IEEE Commun. Surv. Tutorials, vol. 19, no. 4, pp. 2768–2796, 2017, doi: 10.1109/COMST.2017.2749442.

[9] J. Yadav and J. Thakur, "BotEye: Botnet detection technique via traffic flow analysis using machine learning classifiers," PDGC 2020 – 2020

[10] 6th Int. Conf. Parallel, Distrib. Grid Comput., pp. 154–159, 2020, doi: 10.1109/PDGC50313.2020.9315792.

[11] S. Saad et al., "Detecting P2P botnets through network behavior analysis and machine learning," 2011 9th Annu. Int. Conf. Privacy, Secur. Trust. PST 2011, pp. 174–180, 2011, doi: 10.1109/PST.2011.5971980.

[12] U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An adaptive multi-layer botnet detection technique using machine learning classifiers," Appl. Sci., vol. 9, no. 11, 2019, doi: 10.3390/app9112375.

[13] M. Alauthaman, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks," Neural Comput. Appl., vol. 29, no. 11, pp. 991–1004, 2018, doi: 10.1007/s00521-016-2564-5.

[14] S. García, A. Zunino, and M. Campo, "Survey on network-based botnet detection methods," Secur. Commun. Networks, vol. 7, no. 5, pp. 878–903, 2014, doi: https://doi.org/10.1002/sec.800.

[15] H. Li, Z. Chen, R. Spolaor, Q. Yan, C. Zhao, and B. Yang, "DART: Detecting Unseen Malware Variants using Adaptation Regularization Transfer Learning," IEEE Int. Conf. Commun., vol. 2019-May, 2019, doi: 10.1109/ICC.2019.8761598.

[16] M. Zolanvari, M. A. Teixeira, and R. Jain, "Effect of imbalanced datasets on security of industrial IoT using machine learning," 2018 IEEE Int. Conf. Intell. Secur. Informatics, ISI 2018, pp. 112–117, 2018, doi: 10.1109/ISI.2018.8587389.

[17] C. F. T. Pontes, M. M. C. De Souza, J. J. C. Gondim, M. Bishop, and M. A. Marotta, "A New Method for Flow-Based Network Intrusion Detection Using the Inverse Potts Model," IEEE Trans. Netw. Serv. Manag., vol. 18, no. 2, pp. 1125–1136, 2021, doi: 10.1109/TNSM.2021.3075503.

[18] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," Comput. Secur., vol. 45, pp. 100–123, 2014, doi: 10.1016/j.cose.2014.05.011.

[19] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Comput. Secur., vol. 31, no. 3, pp. 357–374, 2012, doi: 10.1016/j.cose.2011.12.012.

[20] A. Alenazi, I. Traore, K. Ganame, and I. Woungang, "Holistic Model for HTTP Botnet Detection Based on DNS Traffic Analysis," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 10618 LNCS, pp. 1–18, 2017, doi: 10.1007/978- 3-319-69155-8 1.

[21] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," ICISSP 2017 - Proc. 3rd Int. Conf. Inf. Syst. Secur. Priv., vol. 2017-Janua, no. Cic, pp. 253–262, 2017, doi: 10.5220/0006105602530262.

[22] M. S. Gadelrab, M. ElSheikh, M. A. Ghoneim, and M. Rashwan, "BotCap: Machine learning approach for botnet detection based on statistical features," Int. J. Commun. Networks Inf. Secur., vol. 10, no. 3, pp. 563–579, 2018.

[23] Bilge, Leyla, et al. "Disclosure: detecting botnet command and control servers through large-scale netflow analysis." Proceedings of the 28th Annual Computer Security Applications Conference. 2012.

ISBN: 978-989-33-3436-2