2020 16th International Conference on Mobility, Sensing and Networking (MSN) | 978-1-7281-9916-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/MSN50589.2020.00087

Early Botnet Detection for the Internet and the Internet of Things by Autonomous Machine Learning

(Invited Paper)

Anderson Bergamini de Neira, Alex Medeiros Araujo, Michele Nogueira
CCSC - Center for Computational Security Science
Federal University of Paraná, Brazil
Emails: {abneira, amaraujo, michele}@inf.ufpr.br

Abstract—**The high costs generated by attacks and the increasing number of different devices on the Internet and the Internet of Things (IoT) propel the early detection of botnets (i.e., network of infected devices) as a way to gain advantage against attacks. However, botnet early detection is challenging due to the continuous mutation, sophistication, and massive data volume, this last mainly resulted from sensor networks and IoT. The literature addresses botnets by modeling the behavior of malware spread, the classification of malicious traffic, and the analysis of traffic anomalies. This paper presents ANTE, a system for ANTicipating botnEts signals based on machine learning algorithms. The ANTE design allows it to adapt to different scenarios by learning to detect different types of botnets throughout its execution. Hence, ANTE autonomously selects the most appropriate machine learning pipeline for each type of botnet to maximize the correct classification before an attack effectively begins. The ANTE evaluation follows a comparison of its results to others from the literature considering three datasets: ISOT HTTP Botnet, CTU-13, and CICDDoS2019. Results show an average accuracy of 99.87% and an average botnet detection precision of 100%.**

1. Introduction

The Internet of Things (IoT) is prominent in academia and industry, with multiple applications. IoT devices have different resource constraints in bandwidth, processing, memory, and energy. This, intensified by vulnerabilities in hardware and software, makes these devices easy targets for attacks. Attackers exploit the vulnerabilities and benefit from the lack of privacy in sensitive data to build massive and damaging attacks. Despite these limitations, the number of Machine-to-Machine (M2M) connections tends to grow. In 2019, CISCO reported the existence of 7.4 billion M2M connections, and the forecast for 2023 is 14.7 billion connections. Also, the number of connected devices is expected to increase, expecting to reach 29.3 billion by 2023. Thus, each inhabitant would have an average of 3.6 connected devices [1], [2]. In 2019, devices connected to the global network transferred around 201 exabytes per month. CISCO predicts that, by 2022, 396 exabytes per month will be generated worldwide [3]. Such growth tends to increase the destructive potential of botnets when exploited by attackers.

A botnet is one of the main problems in cybersecurity. A bot is a device connected to the Internet or IoT and infected by a malware, allowing an attacker to remotely control malicious actions. A botnet comprises infected devices able to execute commands from botmasters (i.e., devices controlled directly by an attacker) [4], [5]. The danger of a botnets comes from the fact that attackers use them to launch different types of attacks. Particularly, botnets serve as a channel to: (i) send bulk electronic correspondence (spam); (ii) capture sensitive information; (iii) and launch Distributed Denial of Service (DDoS) attacks. In addition to the classic uses of botnets, botmasters are always proposing new ways of exploiting botnets [6], and new types of attacks. For instance, the literature highlights that recent botnets disseminate software for cryptocurrency mining [7].

Technological diversity provides the creation of botnets with unique characteristics making the botnet detection a challenge [8]. The geographical distribution and device mobility of bots also increases the difficulty in neutralizing botnets and their effects [5]. Botnets can cover different nations and have regional characteristics [9]. Moreover, Karim et al. [10] claimed that malware hides malicious code so efficiently that several signature-based detection approaches are unable to detect them. With the advent of new technologies and attack sophistication, mainly as result of IoT, botmasters manage to avoid detection techniques. Gupta and Badve [11] mentioned that, in general, attackers manage to flood servers with malicious network packets similar to real ones, making it challenging to differentiate malicious flow from the real one.

Chang et al. [12] found that attackers are alternating between sending packets and remaining temporarily inactive to make bot identification and botnet detection harder. Another serious limitation lies in the fact that it is non-trivial to select the proper detection technique and its best configuration, due to attack diversity, including those unknown (zero-day) attacks. Given the variety of botnet uses, and the difficulty in identifying bots, it is essential that the literature presents solutions to detect botnets. A solution needs to anticipate the botnet formation to prevent further damage to services caused by attacks as well as to adapt themselves to treat each attack [5].

This work presents ANTE, a system for ANTicipating

978-1-7281-9916-0/20/$31.00 ©2020 IEEE
DOI 10.1109/MSN50589.2020.00087

botnEt detection. ANTE maps the behavior of bots with autonomous machine learning techniques creating models to anticipate botnet signals. Different from other works in the literature, such as [13], the ANTE design allows it to adapt to different scenarios, learning to detect different types of botnets throughout its execution. Therefore, ANTE autonomously selects the most appropriate machine learning pipeline for each scenario, to maximize the correct classification of bots before an attack begins. The machine learning pipeline consists of three parts: (i) data collection and preprocessing; (ii) feature preprocessing; (iii) estimator. Machine learning algorithms are relevant for identifying botnets, offering adaptation, and treatment for massive and high-dimensional data. However, each machine learning algorithm has its particularities and not always perform well for the different uses of botnets. Thus, we propose the ANTE system to autonomously decide how to deal with these particularities of the different botnets, obtaining the best use for machine learning techniques.

The ANTE evaluation has as input network traffic captures from three datasets: (i) Information Security and Object Technology (ISOT) HTTP Botnet Dataset, (ii) capture number 51 of Czech Technical University datasets (CTU-13), and (iii) DDoS Evaluation Dataset (CICDDoS2019). The experiments automatically select machine learning algorithms following the ANTE design. Using ANTE, results from the three different scenarios show an average accuracy of 99.87% and an average botnet detection precision of 100%. These results indicate how ANTE can effectively choose an algorithm and the feature preprocessing steps for identifying botnets. Results also show the diversity in the proposed solutions, with a different machine learning pipeline for each scenario, demonstrating that ANTE handles the different types of botnets.

This paper proceeds as follows. Section 2 presents related works. Section 3 details the ANTE system. Section 4 describes the evaluation method and results. Finally, Section 5 concludes the work and highlights future directions.

2. Related Works

Different works in the literature have investigated how to detect infected devices and botnets. These works addressed this problem at both system and network levels. In a nutshell, the main techniques for detecting botnets follow the approaches: (i) anomaly-based detection, (ii) traffic signature-based detection, (iii) graphs, (iv) supervised and unsupervised machine learning.

Anomaly-based detection techniques aim at identifying misbehavior based on the observed characteristics, such as ports used in data traffic, high network latency, or increased traffic volume. In signature-based detection, solutions know previously the botnet behavior and they thus identify the infected devices. Graph-based techniques generate mathematical models to show the relationship between different network objects and detect bots. Entropy maps the degree of randomness of the network and the interception of communication between malicious devices.

This section focuses mainly on machine learning techniques applied to botnet detection. They learn from large data amount, being quick and assertive in automating tasks, such as classification, in different scenarios or even estimating values of continuous variables. Due to the diversity in machine learning techniques, this paper assists in understanding their effectiveness for the early detection of botnets.

The most common machine learning techniques focus on their adaptation to train models able to distinguish data flow from real users to malicious data flows [14]. These techniques have been also applied to anticipate attacks based on bot activities [15], [16]. Supervised machine learning trains models, as it uses a set of labeled data before suggesting to the model how a new record would be classified. There are several supervised machine learning algorithms. The literature often employs neural networks and decision tree-based methods [16], [17], [18]. Another approach groups devices with similar data flow to distinguish legitimate devices from bots [19], [20]. Unsupervised algorithms do not need training before classifying data flows.

In [15], the authors highlighted the different stages of botnet behavior and created a model to predict attacks. Its main limitation is the number of mapped states. If a botnet has a no-mapped state, the solution does not work properly. Unlike [15], for the ANTE system, the infection behavior does not influence the early identification of bots. In [16], the authors focused on identifying botnet command and control sessions (C&C). They analyzed a vector of 55 session characteristics using the Random Forest algorithm to identify C&C. However, their proposal does not identify decentralized botnets, such as peer-to-peer (P2P) ones. In [20], the authors employed a similar approach to group the P2P sessions of the infected devices.

The Autonomous Machine Learning (AutoML) study field aims at democratizing the use of machine learning. Hence, the AutoML automates part of the work with machine learning by automating the selection of techniques and hyperparameters [21]. Different approaches [22], [23] have studied the problem of selecting the best hyperparameters. However, there are recent efforts to unify the choice of machine learning techniques with the best choice of hyperparameters. This problem is called the Combined Algorithm Selection and Hyperparameter (the CASH problem). One of the first approaches in this area is Auto-Weka (2013) [24], followed by other approaches, such as Auto-Sklearn [25] and Auto-Keras [26].

3. The ANTE System

This section describes the system for ANTicipating botnEts signals based on machine learning algorithms (ANTE). The ANTE system captures network traffic and searches for bots to notify network administrators before the effects of the botnet are irreversible, such as a server exhaustion due to a DDoS attack. After capturing network traffic, the system takes out the behavior of the network hosts and, using autonomous machine learning (AutoML), ANTE chooses the preprocessing strategy, the classifier, and the hyperparameters for the classifier. Based on this information, ANTE creates a machine learning pipeline, applying the preprocessing, and it trains the chosen technique to anticipate the beginning of the attack or the botnet activities.

ANTE follows five steps: (i) the capture of network traffic; (ii) the extraction of host behavior; (iii) the identification of the best model (AutoML); (iv) the anticipation of bots; and (v) administrator notification. Figure 1 shows the positioning of the ANTE system in a network. This router must have a port mirroring function so that the network administrator can configure it to forward a copy of the network traffic to the device that is hosting ANTE. Thus, ANTE is able to analyze this data and extract the features that represent the behavior of each active device. ANTE applies autonomous machine learning to identify the best pipeline for the moment. This pipeline comprises actions such as data preprocessing techniques and features that optimize data distribution and choose the most appropriate machine learning model for the moment. After defining the pipeline, the system is ready for being deployed in the real world, where it is able to analyze the behavior of the devices on a network and notify the administrator when any bots are identified. The next subsections detail the relationship between these steps.

3.1. Traffic Capture and Behavior Extraction

The system analyzes network traffic. Thus, as first step, the internal analysis component is responsible for collecting all network traffic and extracting its features. In a real network, the internal analysis component is implemented in either a physical or virtual machine. In situations where there is a massive amount of traffic, it is possible to configure the internal analysis component to use a fraction of the total amount of captured network traffic instead of processing the entire capture. Such action takes place without a significant impact on the overall system performance.

The internal analysis component operates both in a streaming mode for real-time data processing and an offline mode, using historical data as input. ANTE collects all network traffic into the internal analysis component, where the system processes all network traffic is processed following a sliding-window approach. Using this approach, two parameters are critical, window length and sliding interval. The window length lies in the duration of the window. The sliding interval is the frequency in which new network packets are added to the window and old packets expire and are excluded from the computation. Choosing a small window length causes a reduction in computational costs. But, it comes with a decrease in accuracy since a small window can not reliably capture the behavior of a host. ANTE can also increase the speed in which a traffic segment is processed by increasing the sliding interval, but this comes with inaccuracy. These issues mean that the system design and implementation must ponder on the trade-off between computational costs and performance.

After grouping the observed network traffic by time windows, a set of relevant features is extracted and composed by applying a series of aggregate functions over the grouped data. Some features require further grouping of the data. For features based on outgoing or incoming traffic from a host, packets are grouped by either source or destination. Moreover, ANTE uses graph-based features, which involves converting the grouped data to an entirely different data structure. The result of these transformations are then used to build the training dataset.

3.2. Identifying the Best Model (AutoML)

In this step, ANTE employs an AutoML framework to analyze the data in the training window and select the best machine learning pipeline which comprises a combination of preprocessing data, machine learning techniques, and hyperparameters. Some frameworks may not have the preprocessing action, causing the unavailability of this stage. But, it is possible to improve the classification results using preprocessing. Among preprocessors, some options improve data distribution, such as reducing dimensionality, modifying features by joining low-discriminating features, and selecting the most representative ones.

Auto-Sklearn is the applied framework in this work. There are 14 options for preprocessing features [25]. Another preprocessing option is to select features to modify the scale, such as resizing or balancing the elements. The classifier choice can be made in different ways. The design of Auto-Sklearn uses 15 supervised classifiers, such as AdaBoost, Bernoulli Naïve Bayes, and Random Forest [25]. First, Auto-Sklearn extracts meta information from the datasets and compares them with those from a previously analyzed collection. As a result of this analysis, ANTE has a set of classifiers and hyperparameters that Auto-Sklearn uses as a seed for the Bayesian optimization process. At the end of the optimization, the output can be an ensemble of the best machine learning models found or the best model with its hyperparameters. In this work, ANTE employs only the best model, its hyperparameters, and the technique employed for preprocessing. In this paper, we train this model in the training window and we have applied for the anticipation and in other tests.

3.3. Anticipating Bot Detection

For identifying a botnet, a machine learning algorithm must classify one or more devices as bots. The output of this classification returns to ANTE, that checks if some device has been classified as a bot. When ANTE identifies a botnet, it must take an action to avoid denying the available services. The literature addresses some options, such as limiting or blocking access to the device, which can be performed manually or automatically at the firewall. In this work, the envisaged action is to communicate with the network administrators through an email providing a list of Internet Protocol (IP) addresses of possible bots An option for automating incident response is to send a JavaScript Object Notation (JSON) message containing the occurrence data, such as the list of bots and the probability of the device

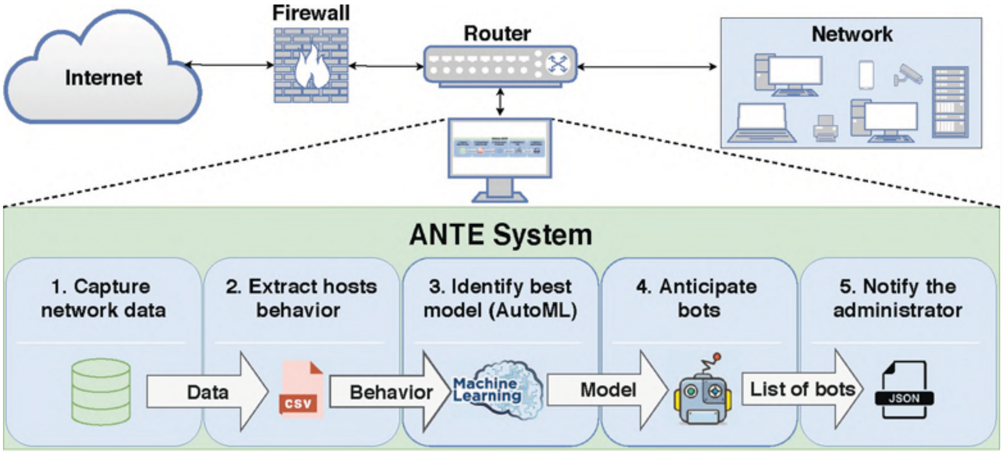

Figure 1: The ANTE Architecture

being a bot. It serves as an entry to automate a firewall or an alert system.

4. Performance Evaluation

Figure 2 illustrates the ANTE evaluation methodology. Evaluations employ as input three datasets where there is data flows from infected (bots) and non-infected (not bots) devices. The three datasets are the (i) *Information Security and Object Technology* (ISOT) *Hypertext Transfer Protocol* (HTTP) Botnet Dataset; (ii) the capture number 51 of the Czech Technical University datasets (CTU-13); and (iii) DDoS Evaluation Dataset (CICDDoS2019). This section details these datasets and all tests that have been conducted using a computer with an I5 processor, 1 terabyte of hard disk drive, and 8 gigabytes (GB) of random-access memory.

As illustrated in Figure 2, having as input the datasets (item 1), ANTE extracts the behavior of the network hosts (item 2). Extraction is necessary for defining the analysis center (i.e., the time when the attack starts). The dataset documentation provides details regarding the actions of the botnets in the network traffic captures, indicating which time the attack begins. After defining this analysis center, ANTE sets the training, anticipation, and testing windows (represented in Figure 2, item 2).

The ANTE system extracts three minutes immediately before the identified actions of the botnets (analysis center). ANTE splits these three minutes of capture into two parts. The first part has two minutes and 30 seconds, and it is named training window. The second part comprises 30 seconds of dataset capture, and it is named anticipation window. After the analysis center, ANTE extracts two more minutes for the identification tests during the attack. ANTE defines the anticipation window and the test window to analyze different stages of the botnet life cycle.

ANTE generates comma-separated values (CSV) extension files for all time windows. These CSVs contain data related to devices and their packet exchange on the network representing the behavior of the devices grouped

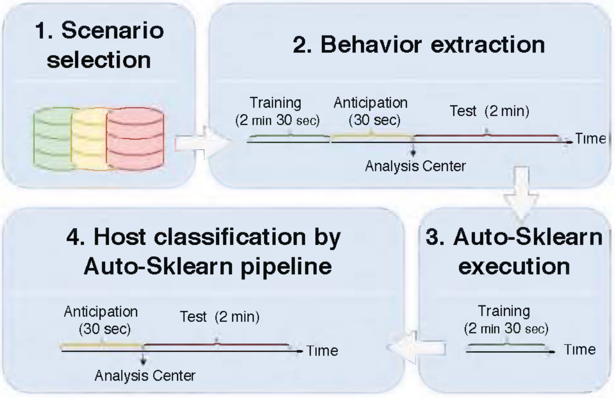

Figure 2: The ANTE System Evaluation Steps

by IP address. Another alternative is to use a combination of IP and Medium Access Control (MAC) addresses. This behavior consists in a combination of 18 flow features [16] and four graph features [27], detailed in Table 1. Therefore, each CSV line represents a different device by the IP and label attributes. The ANTE system extracts all the features present in the CSV (Table 1) from all datasets. Thus, the system does not favor features and ensures that ANTE can use all features for all analyses. In order to ensure the training of the supervised learning, we provide to ANTE the malicious device identifier (label) from the documentation for the respective databases. Also, ANTE only employs the IP and label to characterize the behavior, and the system does not use them as a flow feature.

Auto-Sklearn operates only with data from the training window (represented in Figure 2, item 3). For the definition of the training data, ANTE uses a sliding window approach where the system extracts the features at evenly spaced fixed-size windows over the training segment. The sliding window size is of 30 seconds and offsets the start of every window by 500 milliseconds. This procedure helps Auto-Sklearn to find the best combination between preprocessors, machine learning models, and hyperparameters for that win-

Features	Description
Tsr/Tss	Sum of received/sent packets size (bytes)
Rpc/Spc	Amount of packets received/sent
Savg	Average sent packet size
Smin	Minimum sent packet size (bytes)
Smax	Maximum sent packet size (bytes)
Svar	Variance of sent packets (bytes)
Ravg	Average received packet size
Rmin	Minimum received packet size
Rmax	Maximum received packet size (bytes)
Rvar	Variance of received packets (bytes)
SITmin	Minimum interval time in sent packages
SITmax	Maximum interval time in sent packages
SITavg	Average interval time of sent packages
RITmin	Minimum interval time in received packages
RITmax	Maximum interval time in received packages
RITavg	Average interval time of received packages
OutDgr	Number of edges pointing out of the node
InDgr	Number of edges pointing in to the node
BC	Number of shortest paths passing by a node
EG	Centrality degree of a node

TABLE 1: Employed Features to Represent Devices

dow. The configuration of Auto-Sklearn is almost standard, except for the processing time configured to 5 minutes, instead of 30 minutes as recommended. Auto-Sklearn selects only the best machine learning model. In order to keep the operating system stable and not hinder testing, Auto-Sklearn uses three of the four processor cores. Finally, to avoid overfitting and benefit models that are able to classify samples from bots and legitimate devices correctly, F1-Score is the metric employed to lead Auto-Sklearn to give priority to equitable models.

ANTE defines the machine learning pipeline for each scenario after the analysis of the training data is completed. The machine learning pipeline consists of three strategies to improve training data and a strategy for classifying devices. The first strategy solves the missing data problem, and we call it the imputation strategy. Missing data is the loss of information that some devices could have for some features because of different reasons, such as transmission issues, failures in collecting data and other. In this evaluation, all the analyzed devices have always the attributes properly estimated. This means that there are no missing data in the dataset. However, we have designed ANTE to propose an imputation strategy able of dealing with missing data, if it occurs while using the system. The second strategy for improving training data is rescaling. Some machine learning algorithms show better results if the data is represented in certain scales, such as representing the values of the attributes in the range between zero and one. Thus, ANTE selects a rescaling strategy for all scenarios. The last strategy for improving training data is preprocessing features. Attribute selection seeks to remove low-discriminating attributes and improve the overall functioning of ANTE. Finally, the last strategy in the pipeline is the estimator. Estimator defines a machine learning technique that will best classify the new devices.

For evaluations, represented in Figure 2, item 4, the ANTE system employs the preprocessing technique, the machine learning model and the hyperparameters selected by Auto-Sklearn, to classify hosts before the attack effectively starts (anticipation window) and throughout the attack (test window). For the creation of the test data, ANTE uses a sliding window size of 30 seconds and offsets the start of every window by 30 seconds. In order to measure the quality of the ANTE decisions, results compare the classification to the original labels using metrics such as accuracy, precision, recall, and the F1-Score, as detailed next. Additionally, metrics evaluate the diversity in the choices of preprocessors, machine learning models, and hyperparameters. Also, evaluations test Support Vector Machines (SVM), Naïve Bayes (NB), and Multilayer Perceptron (MLP) in all scenarios. This comparison analyzes the importance of the correct choice and configuration of machine learning techniques and to examine whether other techniques would be assertive. Finally, evaluation scenarios also compare ANTE results to the results in [28].

The first employed metric is accuracy (Eq. 1), which relates the total number of correct classifications to the total of classified items. However, in cases where the distribution of classes is unbalanced, this metric may be misinterpreted. Due to the classifier can classify all predictions for the majority of classes correctly and makes mistakes for all projections to the minority class (and still has accuracy above 90%). Precision (Eq. 2), recall (Eq. 3), and F1-Score (Eq. 4) are metrics based on the number of true positives (TP), false positives (FP), false negatives (FN), true negatives (TN). Hence, it is possible to obtain these metrics for the class of non-bots and bots. Thus, for this work, the bot collection is the positive class and present the precision, recall, and F1-Score for the bot class. All information is available in the project repository[1].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 - Score = 2. \frac{precision \, . \, recall}{precision + recall} \tag{4}$$

4.1. Results

The **ISOT Scenario** is a combination of several malicious and non-malicious datasets. The University of Victoria

[1] https://github.com/axms/ANTE

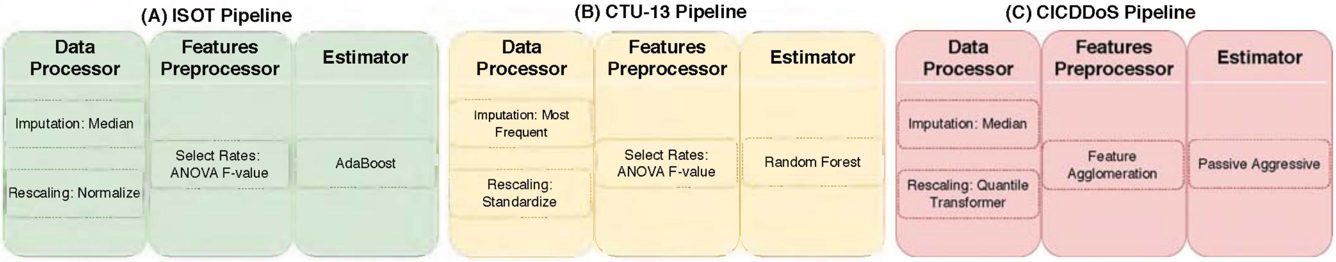

Figure 3: The Best ML Pipeline per Scenario Identified by ANTE

in Canada[2] is the maintainer of this dataset. The scenario used in our evaluation consists of malicious and benign Domain Name System (DNS) traffic, totaling 780 megabytes of network traffic data generated by 35 devices, nine of them are infected. The analysis center was defined as 21:39:31 on 30 May 2017 Coordinated Universal Time (UTC), since the goal of the attack was the information theft. ANTE trains with the presence of eight bots among the 35 devices that have exchanged packets before the beginning of the attack.

Figure 3A illustrates the entire configuration of the machine learning pipeline chosen by ANTE after analyzing the training data for the ISOT scenario. Further information about the configurations is public in the repository[3]. ANTE chooses the Median as the most appropriate imputation strategy for this scenario. This means that if during the system execution the feature extraction component cannot calculate any attribute for any device, ANTE completes it using the median of the value of the same attribute of the other devices. ANTE has also identified that changing the data scale by applying attribute normalization would be beneficial for the device classification. Thus, ANTE normalized the data on a norm scale. The last strategy to improve training data lies in applying the Select Rates approach. *Select Rates* is an approach to feature selection based on statistical tests. *Select Rates* can follow three strategies to measure the quality of attributes, thus keeping the most discriminating ones. The selected strategy was the ANOVA F-value between the label and the features. Finally, ANTE selects the AdaBoost estimator as the most appropriate machine learning technique for the scenario.

Table 2 shows the result of applying the entire machine learning pipeline selected by ANTE. Using the ANTE pipeline, we obtained the correct classification for all devices in the step before the beginning of the attack. But, the algorithm misclassified a bot by marking it as a regular device in the test window. Also in the Table 2, we see the difficulty that SVM, NB, and MLP have to correctly classify devices, as they were not as effective as the ANTE approach.

The **CTU-13 Scenario** uses the capture-51 belonging to the set of 13 captures, called CTU-13 [29]. This set of 13 captures of botnet traffic was recorded at the Czech Technical University made available through the Stratosphere Project[4]. This dataset has about 66 GB and reports two attacks. In this scenario, we analyze the second attack that starts on August 18, 2011, at 14:43:27 Central European Summer Time. This scenario contains an ICMP Flood Attack using 10 bots. During the training, ANTE has identified 30 devices of which 10 were bots.

Approach	Window	Accuracy	Precision	Recall	F1-Score
ANTE	Anticipation	100%	100%	100%	100%
	Test	99.23%	100%	96%	98%
SVM	Anticipation	100%	100%	100%	100%
	Test	93%	100%	62%	77%
Naïve Bayes	Anticipation	72%	50%	100%	67%
	Test	74%	40%	83%	54%
MLP	Anticipation	96%	100%	86%	92%
	Test	94%	94%	71%	81%

TABLE 2: Results for the ISOT Scenario

Figure 3B illustrates the entire configuration of the machine learning pipeline chosen by ANTE. Unlike the previous scenario, the ANTE chooses the Most Frequent as the imputation strategy and the *Standardize* approach as the rescaling strategy. If ANTE identifies missing values, it completes them using the most common value of the feature. The *Standardize* strategy standardizes features by applying a process where ANTE calculates the average and standard deviation for each feature across the training base. For each new observation (new device to be classified), ANTE removes the average value and divides it by the previously calculated standard deviation for each feature. As in the previous scenario, ANTE chooses the *Select Rates* approach as a feature processor. But the configuration of the hyperparameters selected by ANTE was slightly different. All pipeline configurations are available in the repository[3]. ANTE chooses a different estimator from the previous scenario. In this case, ANTE identifies the *Random Forest* estimator as the best option.

ANTE preprocesses the data and classifies the devices present in the anticipation and test windows using the entire pipeline that is present in Figure 3B. Although the ANTE of the pipeline of this scenario has differences in the pipeline of the previous scenario. The Table 3 shows that the model was able to classify all devices correctly in the two windows. A similar characteristic to the previous scenario was the difficulty of the estimators SVM, NB, and MLP to obtain to correctly classify the hosts.

The **CICDDoS Scenario** uses the CICDDoS2019

[2] https://www.uvic.ca/engineering/ece/isot/datasets/botnet-ransomware Accessed in: 08/2020

[3] https://github.com/axms/ANTE

[4] www.stratosphereips.org Accessed in: 08/2020

Approach	Window	Accuracy	Precision	Recall	F1-Score
ANTE	Anticipation	100%	100%	100%	100%
	Test	100%	100%	100%	100%
SVM	Anticipation	92%	50%	100%	67%
	Test	74%	69%	100%	82%
Naïve Bayes	Anticipation	92%	0%	0%	0%
	Test	100%	100%	100%	100%
MLP	Anticipation	92%	50%	100%	67%
	Test	97%	95%	100%	98%

TABLE 3: Results for the CTU-13 Scenario

dataset [30], made available by the University of New Brunswick (UNB)[5]. We choose this dataset because it was published in 2019, it has 20 GB of network traffic, and it has different types of attacks collected in two days [30]. Of the 12 attacks reported in this collection, this scenario analyzes the Network Basic Input Output System (NetBIOS) attack conducted on the first day of the experiment. According to the documentation, this attack took place between 10:21 and 10:30 on the first day. However, analyzing the available data, it is not possible to find network flow at this time. Thus, the tests have investigated the peak traffic that occurred on December 1, 2018 at 15:06:10 UTC as the analysis center. Examining the data of this scenario, there is a peak of 110 devices. However, it has only one active bot.

After the analysis of the training data by ANTE, the machine learning pipeline selected has also a difference when we compare it with the previous scenarios. Figure 3C indicates the choices made by ANTE to compose the pipeline. The full set of settings is available in the repository[3]. About data processors, the imputation strategy chosen for this scenario was the same as that of the ISOT a median scenario, however the *Quantile Transform* strategy for rescaling is different from the three scenarios. In practice, this Quantile Transform tends to spread the most frequent values among the features because the purpose of this method is to transform the data into a normal distribution. ANTE chose a feature preprocessing technique different from the previous scenarios. Feature Agglomeration reduces the dimensionality of features (number of features) by grouping similar features. Finally, ANTE has chosen *Passive Aggressive* as the estimator, which is also different from previous scenarios.

Table 4 shows the results obtained using the pipeline defined by ANTE. It is possible to verify that the proposed approach correctly classifies all hosts for the anticipation and the test windows. As observed in other scenarios, the SVM, NB, and MLP estimators were not successful in classifying the devices found in the scenario.

Approach	Window	Accuracy	Precision	Recall	F1-Score
ANTE	Anticipation	100%	100%	100%	100%
	Test	100%	100%	100%	100%
SVM	Anticipation	98%	0%	0%	0%
	Test	99%	0%	0%	0%
Naïve Bayes	Anticipation	98%	50%	100%	67%
	Test	100%	100%	100%	100%
MLP	Anticipation	94%	25%	100%	40%
	Test	98%	44%	100%	62%

TABLE 4: Results for the CICDDoS Scenario

4.2. Comparison with the literature

For comparative purposes, we have reproduced the results obtained in [28] using the CTU-13 dataset. Also, we add the analysis of the ISOT HTTP and CICDDoS2019 datasets. The first comparison uses the obtained results by the technique proposed in [28] on the ISOT HTTP Botnet Dataset. In this dataset, Pektaş and Acarman's solution got all the classifications right as well as the ANTE system (anticipation window). In the CTU-13 dataset, the ANTE approach is superior to Pektaş and Acarman's one. ANTE is able to classify all hosts correctly, while Pektaş and Acarman's approach has reached an accuracy of 99.3%. The last comparison is to the CICDDOS2019 dataset. For it, Pektaş and Acarman's approach obtains an accuracy of 96.87%. The issue with this result is that, throughout the tests, the proposed method fails to identify the bots. ANTE is able to classify all devices correctly. Table 5 provides all data referring to the comparison.

Dataset	System	Accuracy	Precision	Recall	F1
ISOT	ANTE	100%	100%	100%	100%
	Ref. [31]	100%	100%	100%	100%
CTU-13	ANTE	100%	100%	100%	100%
	Ref. [31]	99.3%	99.1%	99.2%	99.1%
CICDDoS	ANTE	100%	100%	100%	100%
	Ref. [31]	96.8%	0%	0%	0%

TABLE 5: ANTE vs. Results from the Literature

5. Conclusion

Given the destructive potential of botnets, the diversity of attacks, and the difficulty in detecting them mainly with the advent of the Internet of Things (IoT) and mobile devices, this paper introduced the ANTE system. Its goal lies in early detecting botnets by autonomous machine learning. For the three analyzed scenarios, the proposed system was assertive in classifying the devices before and during the actions of the evaluated botnets. For the system to achieve this result, autonomous machine learning selects the best machine learning pipeline for each scenario. This indicates that ANTE takes into account the particularities of each scenario, including different types of attacks, and different botnets while choosing the most suitable techniques for each scenario. Thus, ANTE has highlighted the importance of the correct choice of techniques to ensure high assertiveness in botnet identification. Another important finding is that, in general, bots are present and have already started exchanging packets even before the effective start of the attacks. Even without the attack being started, ANTE was able to identify bots in advance. Future works include the reduction in the dependence on labels using other strategies. Also, they involve the improvement in the processing time of ANTE to make the solution feasible online.

Acknowledgment

This work was supported by National Council for Scientific and Technological Development (CNPq/Brazil), grants

[5] www.unb.ca/cic/datasets/ddos-2019.html Accessed in: 08/2020

#309129/2017-6 and #432204/2018-0, by São Paulo Research Foundation (FAPESP), grant #2018/23098-0, by the Coordination for the Improvement of Higher Education Personnel (CAPES/Brazil), grants #88887.501287/2020-00 and #88887.509309/2020-00, and by the National Teaching and Research Network (RNP) by the GT-Periscope project.

References

[1] A. Nordrum, "Popular internet of things forecast of 50 billion devices by 2020 is outdated (2016)," *https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated [Accessed in: 08/2020]*, 2016.

[2] V. N. I. Cisco, "Cisco annual internet report (2018–2023) white paper," *https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html [Accessed in: 08/2020]*, 2020.

[3] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco visual networking index (vni), complete forecast update, 2017–2022," *Americas/EMEAR Cisco Knowledge Network (CKN) Presentation*, 2018.

[4] Y. D. Mane, "Detect and deactivate p2p zeus bot," in *ICCCNT*, July 2017, pp. 1–7.

[5] A. A. Santos, M. Nogueira, and J. M. F. Moura, "A stochastic adaptive model to explore mobile botnet dynamics," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 753–756, 2017.

[6] S. Venkatesan, M. Albanese, A. Shah, R. Ganesan, and S. Jajodia, "Detecting stealthy botnets in a resource-constrained environment using reinforcement learning," in *Proceedings of the 2017 Workshop on Moving Target Defense*, ser. MTD '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 75–85. [Online]. Available: https://doi.org/10.1145/3140549.3140552

[7] D. Y. Huang, H. Dharmdasani, S. Meiklejohn, V. Dave, C. Grier, D. Mccoy, S. Savage, N. Weaver, A. C. Snoeren, and K. Levchenko, "Botcoin: Monetizing stolen cycles."

[8] S. Ruano Rincón, S. Vaton, A. Beugnard, and S. Garlatti, "Semantics based analysis of botnet activity from heterogeneous data sources," in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015, pp. 391–396.

[9] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," *Computer Networks*, vol. 57, no. 2, pp. 378 – 403, 2013, botnet Activity: Analysis, Detection and Shutdown. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128612003568

[10] A. Karim, R. B. Salleh, M. Shiraz, S. A. A. Shah, I. Awan, and N. B. Anuar, "Botnet detection techniques: review, future trends, and issues," *Journal of Zhejiang University SCIENCE C*, vol. 15, no. 11, pp. 943–983, Nov 2014. [Online]. Available: https://doi.org/10.1631/jzus.C1300242

[11] B. Gupta and O. P. Badve, "Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment," *NCA*, vol. 28, no. 12, pp. 3655–3682, 2017.

[12] W. Chang, A. Mohaisen, A. Wang, and S. Chen, "Understanding adversarial strategies from bot recruitment to scheduling," in *SecureComm*, X. Lin, A. Ghorbani, K. Ren, S. Zhu, and A. Zhang, Eds. Cham: SIP, 2018, pp. 397–417.

[13] B. M. Rahal, A. Santos, and M. Nogueira, "A distributed architecture for ddos prediction and bot detection," *IEEE Access*, vol. (Early Access), no. 1, pp. 1–17, 2020.

[14] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, and P. Hakimian, "Detecting P2P botnets through network behavior analysis and machine learning," in *PST*, July 2011, pp. 174–180.

[15] Z. Abaid, D. Sarkar, M. A. Kaafar, and S. Jha, "The early bird gets the botnet: A markov chain based early warning system for botnet attacks," in *IEEE on LCN*, Nov 2016, pp. 61–68.

[16] L. Lu, Y. Feng, and K. Sakurai, "C&c session detection using random forest," in *IMCOM*. New York, NY, USA: ACM, 2017, pp. 34:1–34:6. [Online]. Available: http://doi.acm.org/10.1145/3022227.3022260

[17] A. Bansal and S. Mahapatra, "A comparative analysis of machine learning techniques for botnet detection," in *SINCONF*. New York, NY, USA: ACM, 2017, pp. 91–98. [Online]. Available: http://doi.acm.org/10.1145/3136825.3136874

[18] S. Chen, Y. Chen, and W. Tzeng, "Effective botnet detection through neural networks on convolutional features," in *IEEE TrustCom*, Aug 2018, pp. 372–378.

[19] S.-H. Li, Y.-C. Kao, Z.-C. Zhang, Y.-P. Chuang, and D. C. Yen, "A network behavior-based botnet detection mechanism using PSO and K-means," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 1, pp. 3:1–3:30, Apr. 2015. [Online]. Available: http://doi.acm.org/10.1145/2676869

[20] C.-Y. Wang, C.-L. Ou, Y.-E. Zhang, F.-M. Cho, P.-H. Chen, J.-B. Chang, and C.-K. Shieh, "Botcluster: A session-based P2P botnet clustering system on netflow," *Computer Networks*, vol. 145, pp. 175 – 189, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618308351

[21] J. G. Madrid, H. J. Escalante, E. F. Morales, W.-W. Tu, Y. Yu, L. Sun-Hosoya, I. Guyon, and M. Sebag, "Towards automl in the presence of drift: first results," *arXiv preprint arXiv:1907.10772*, 2019.

[22] N. Lu, J. Zhou, Y. He, and Y. Liu, "Particle swarm optimization for parameter optimization of support vector machine model," in *ICICTA*, vol. 1, 2009, pp. 283–286.

[23] Jingming Peng and Shuzhou Wang, "Parameter selection of support vector machine based on chaotic particle swarm optimization algorithm," in *ICA*, 2010, pp. 3271–3274.

[24] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka," *J. Mach. Learn. Res.*, vol. 18, no. 1, p. 826–830, Jan. 2017.

[25] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *NIPS*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2962–2970. [Online]. Available: http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf

[26] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD*. ACM, 2019, pp. 1946–1956.

[27] S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H. Medal, M. Marufuzzaman, and L. Bian, "Botnet detection using graph-based feature clustering," *Journal of Big Data*, vol. 4, p. 14, 04 2017.

[28] A. Pektaş and T. Acarman, "Deep learning to detect botnet via network flow summaries," *NCA*, vol. 31, no. 11, pp. 8021–8033, 2019.

[29] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *C&S*, vol. 45, pp. 100 – 123, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404814000923

[30] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *ICCST*, Oct 2019, pp. 1–8.

[31] A. Pektaş and T. Acarman, "Deep learning to detect botnet via network flow summaries," *NCA*, Jul 2018. [Online]. Available: https://doi.org/10.1007/s00521-018-3595-x