Early Identification of Abused Domains in TLD through Passive DNS Applying Machine Learning Techniques

Leandro Marcos da Silva[1], Marcos Rogério Silveira[1], Adriano Mauro Cansian[1] and Hugo Koji Kobayashi[2]

[1]Sao Paulo State University (UNESP), Department of Computer Science and Statistics (DCCE), Brazil
[2]Brazilian Network Information Center (NIC.br), Brazil

***Abstract*:** DNS is vital for the proper functioning of the Internet. However, users use this structure for domain registration and abuse. These domains are used as tools for these users to carry out the most varied attacks. Thus, early detection of abused domains prevents more people from falling into scams. In this work, an approach for identifying abused domains was developed using passive DNS collected from an authoritative DNS server TLD along with the data enriched through geolocation, thus enabling a global view of the domains. Therefore, the system monitors the domain's first seven days of life after its first DNS query, in which two behavior checks are performed, the first with three days and the second with seven days. The generated models apply the machine learning algorithm LightGBM, and because of the unbalanced data, the combination of Cluster Centroids and K-Means SMOTE techniques were used. As a result, it obtained an average AUC of 0.9673 for the three-day model and an average AUC of 0.9674 for the seven-day model. Finally, the validation of three and seven days in a test environment reached a TPR of 0.8656 and 0.8682, respectively. It was noted that the system has a satisfactory performance for the early identification of abused domains and the importance of a TLD to identify these domains.

***Keywords*:** cybersecurity, passive DNS, abused domains in TLD, data imbalanced, machine learning algorithms.

1. Introduction

Malicious or abused domains are used as tools by malicious users on the Internet, whose purpose is usually for use in phishing, malware, Command and Control (C&C), and fast-flux domains. More than 566 million registered domains worldwide, distributed over 1,599 Top-Level Domains (TLDs) [1]. Due to this vast quantity of domains, the Domain Name System (DNS) becomes an essential component for the proper functioning of the Internet. DNS performs the translation of domain names into Internet Protocol (IP) addresses and vice versa [2].

For domain name resolution, DNS has a hierarchical structure similar to a tree. Thus, when a user wants to access a website, the resolver requests a recursive server to start the domain name resolution process. The resolver starts the resolution process by sending a request to the root server and getting IP addresses from the DNS TLDs servers in response. Server TLDs can be divided into the Country Code Top-Level Domain (ccTLD) and Generic Top-Level Domain (gTLD). ccTLDs are designated for countries, so it depends on their geographic location, and gTLDs are generic in use without geographic issues. Afterward, the resolver sends the request to the TLD server responsible for the domain it is resolving and gets the IP of a Second-Level Domain server in response, and so on, until the resolution process is complete and has the server's IP domain authoritative. Once such information is obtained, the recursive server sends it to the resolver [3].

When a DNS server responds to a request, one or more Resource Records (RRs) are sent. Furthermore, it is possible to perform the DNS query requesting specific RRs. The RR can be understood as a tuple with four fields: name, value, type, and Time to Live (TTL) [4]. The name and value fields depend on which type of RR is requested. The TTL field indicates how long that response can be cached on the DNS server. The most common type fields are typed A and AAAA, which suggests that the answer is the definition of the IPv4 and IPv6 addresses, respectively. Also, there is CNAME, which defines an alias for a Fully Qualified Domain (FQDN).

In registering a domain, the user accesses a registrar responsible for selling a specific domain name, enters with his personal information, the domain he wants to acquire, and the data to effectuate payment. This step is understood as pre-registration. Authors who choose to identify abused domains in this step [5] use lexical features coming from the domain and the registrant's data, which can prevent the abused domain from being used. After the payment and the first update of the DNS zone, the authors usually employ the approaches through the active [6] and passive [7] DNS data, called the post-registration.

Passive DNS is the collection of communication between DNS servers, performed by sniffers installed in the network, in which it is possible to obtain queries and responses from DNS servers [8]. Passive DNS can be collected at different points in the DNS structure process, the most common being recursive servers or authoritative TLD servers. In this work, the passive DNS collected from a TLD was chosen for its global domain visibility [9].

Looking at the proportion of users attacked by phishing in each country, Brazil had the highest number, accounting for 6.63% of all triggers based on Kaspersky's report [10]. Also, according to Symantec's Internet Security Threat Report (ISTR) [11], one in ten URLs is considered malicious. Thus, early identifying abused domains is necessary, preventing more Internet users from falling into scams or their personal computers from being compromised. As a result, there are fewer victims and minor damage, where damages range from financial to data theft.

Due to the high number of domain names and requests that DNS servers receive, automated detection approaches that apply Machine Learning (ML) techniques are increasingly helpful in combating abused domains. With the passive DNS of an authoritative TLD server, it is possible to track the domain from the beginning, after its first zone update. An inherent problem with this approach is that the number of legitimate domains is much higher than the number of abused domains, resulting in a highly unbalanced dataset presented in Section III. Thus, it is necessary to use techniques to balance

the data so that the developed model does not have a data bias. Regarding data balancing techniques, they are divided into three families: undersampling, which works by removing data from the majority class; oversampling, which generates replicas or synthetic data from the minority class; and hybrids, in which the two previous families are combined [12]. There are several undersampling techniques, including Random Undersampling (RUS), which equalizes data by randomly removing data from the majority class; and Cluster Centroids (CC), which uses K-Means to locate the centroids of clusters, containing the majority class data, in addition to excluding data that is far from the centroid [13]. About the oversampling techniques, the most applied are Random Oversampling (ROS), which equalizes through the random replication of data from the minority class; and the Synthetic Minority Oversampling Technique (SMOTE), which uses the K-Nearest Neighbors (KNN) to generate the synthetic data between neighbors [14]. There are also techniques derived from SMOTE, such as Borderline-SMOTE [15], SVM-SMOTE [16], and K-Means SMOTE [17].

This work presents an approach to early identification of abused domains, in which the domain is monitored in the first three and seven days of life after the first DNS request, thus ensuring that a registered and unused domain is monitored for this period after the actual start of its use. The passive DNS used to detect these abused domains is provided by an authoritative TLD server, collected during 12 months. There is an enrichment of the data in the collected passive DNS through the ENTRADA [18]. The presented approach uses ML algorithms prepared to support the massive amount of data. As contributions to this work, there are:

- Early and automated identification of domains that are malicious behavior based on their DNS traffic in the TLD;
- Monitoring the domain's first seven days of life after its first DNS query;
- An approach that makes two checks on the behavior of the domain, one with three days and the other with seven days;
- Application of data balancing techniques and ML algorithms are easily scalable and ready to handle large amounts of data. Bayesian optimization is used to obtain the best hyperparameters for the Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting (LightGBM) algorithms.

The present work is divided into five sections, wherein Section 1, the introduction of the work is presented; in Section 2, the related works are discussed; in Section 3, the methods of the system are presented, from the collection of passive DNS to the preparation and testing of models in production; in Section 4 the results obtained with the development of the models are presented, as well as a discussion; and, finally, in Section 5 the conclusions and future works.

2. Related Works

Several approaches work in the post-registration stage to detect malicious domains using passive DNS as the data source. The distinction is due to the ML techniques, the features employed, and the collected servers, which can be recursive or authoritative. Thus, in the sequence, works related to the system defined in this work will be presented.

The first dynamic reputation system presented was Notos [19] in 2010. The authors assume that a legitimate domain would receive a higher score while malicious domains receive a low score. The Decision Tree (DT) algorithm trained on a dataset was used in the system, and after the training stage, Notos is executed online.

Introduced in 2011, Exposure [20] uses passive DNS collected from a recursive server to extract its features based on time, DNS response, TTL, and others based on the domain name. The classifier model applies the DT algorithm to detect domains related to spam, malware, fast-flux domains, and Domain Generation Algorithms (DGAs).

Unlike the works presented above, Kopis [9] uses passive DNS collected from the ".ca" TLD and authoritative servers, which provides it with global visibility. Kopis is designed to detect malware-related malicious domains, in which the collected passive DNS is separated into epochs, with each collected day being considered an epoch. The features used by the classifier model are computed daily, and the chosen algorithm was Random Forest (RF).

Opting for a neural approach, Lison *et al*. [21] use two neural networks, the first being a recurrent neural network that receives the domain name, thus calculating the probability that the domain was generated by malware. This result, together with the other extracted features, is the input to a feedforward neural network, in which the domain can be classified as legitimate, malicious, or sinkhole. The authors used passive DNS for feature extraction.

Using the XGBoost algorithm, Bao *et al*. [22] present an approach to detect malicious domains related to DGAs and the detection of domains linked to pornography based on the word vector. The authors collected 3-day passive DNS from the Chinese environment and cited the data imbalance problem, with the ratio being 1:20, and applying the downsampling technique, reducing the data imbalance to 1:5.

Wang *et al*. [23] present a method for dataset resampling, in which the K-Means SMOTE oversampling technique is used to balance the data. In the approach used, the authors combine passive and active DNS data to detect malicious domains using the Categorical Boosting (CatBoost) algorithm because there is an appropriate number of categorical features.

Finally, in Silveira *et al*. [7], the XGBoost algorithm was used to detect malicious domains. The authors use the passive DNS as their only data source, where all extracted features come from the passive DNS. To solve the problem of unbalanced data, they chose to use the RUS technique. For selecting the best hyperparameters in XGBoost, Bayesian optimization with 21 initial points and 300 iterations was applied.

Given the above, several authors use passive DNS to detect abused domains, which differentiates the works in relation to passive DNS is the source from which it is collected, where it is highlighted that only Kopis [9], which makes use of passive DNS collected from an authoritative TLD server, which resembles the data source used in this work. It is possible to observe that in only three works, the unbalance of the training dataset is treated, in which one has used a downsampling technique [22], in Wang *et al*. the K-Means SMOTE [23] was used, and in Silveira *et al*. [7] used the RUS technique. In this work, a combination of undersampling and oversampling techniques is used and compared. Finally, it is possible to observe from related works, except for Lison *et al*. [21], that the authors choose to work with algorithms based on DT, and only three use algorithms with the boosting method, being XGBoost [22,7] and CatBoost [23], and only Silveira *et al*. [22] used Bayesian optimization to improve the algorithm's performance. In this work, an algorithm based on trees and

with boosting is used, different from the ones applied so far, LightGBM, with similar performance to XGBoost, but with a much shorter training time. Finally, Bayesian optimization is also used to select the model's hyperparameters and improve the results.

3. Methods

In this section, the methodology of the developed system will be presented. Its role is to detect newly registered abused domains through monitoring in the first week of use. To achieve this goal, it is necessary to apply ML techniques to build the models and use the passive DNS dataset. The system is divided into two parts: preparation of models, being responsible for generating the two models that monitor the newly registered domains in the first three and seven days after the first query; and models in production, which is in charge of putting previously built models into operation.
Based on the explanations above, a diagram was created with the system overview presented in Figure 1, and in which each subsection, the diagram is explained in detail.

3.1 Passive DNS

As seen in Figure 1, the passive DNS dataset is used, which is formed from the collection of DNS traffic from an authoritative TLD server through passive DNS. The system responsible for the collection uses ENTRADA [18] instances, which clean, enrich and compress the data and insert it into a Hadoop Cluster. ENTRADA is a tool that allows the analysis of large amounts of DNS data in seconds to very few minutes. Initially, the dataset contains all the ENTRADA columns exclusive to passive DNS, except for the country codes and Autonomous System Number (ASN), generated from data enrichment with the help of the GeoIP database of Maxmind. Therefore, the generated dataset comprises 12 months of DNS traffic collection, starting in early March 2020.

3.2 Data Filtering and Selection

After generating the passive DNS dataset, the data were filtered based on a list of registered domains provided by the team responsible for registering domains in the TLD collected. The list contains all domains registered from the beginning of March to the end of December 2020, thus offering 1,348,938 domains. After filtering the previously generated dataset, 1,304,893 queries from newly registered domains resulted. It is interesting to note that the collection period is two months longer than the list because domains tend to be registered and take until the first query. After filtering the data, the selection of data from X days is made, where the value X corresponds to the period of three or seven initial days of the domain's life. It is noteworthy that each X is equivalent to a model to monitor the domains, and as the X can be three or seven, the system is composed of two models.

3.3 Feature Extraction

In the extraction of features, the queries of the newly registered domains are added, and the features are extracted, with the features being equivalent to periods of X days. Due to the purpose of this work being to identify abused newly registered domains in a TLD and use the ENTRADA, there is the presence of exclusive features marked with an asterisk (*) beside. However, most of the features were chosen based on previous works that address the use of passive DNS to detect malicious domains [21,24,25]. Table 1 shows all the features used in the proposed system in this work.
Table 1 shows the extraction of 20 features exclusively from the passive DNS and the features provided by the columns enriched in ENTRADA. It also highlights the use of only numeric features, where the lexical features were used only to help in data labeling, such as the domain name, and not considered for training. As a result of the aggregation of queries, 89,988 domains were obtained referring to data from three and seven days.

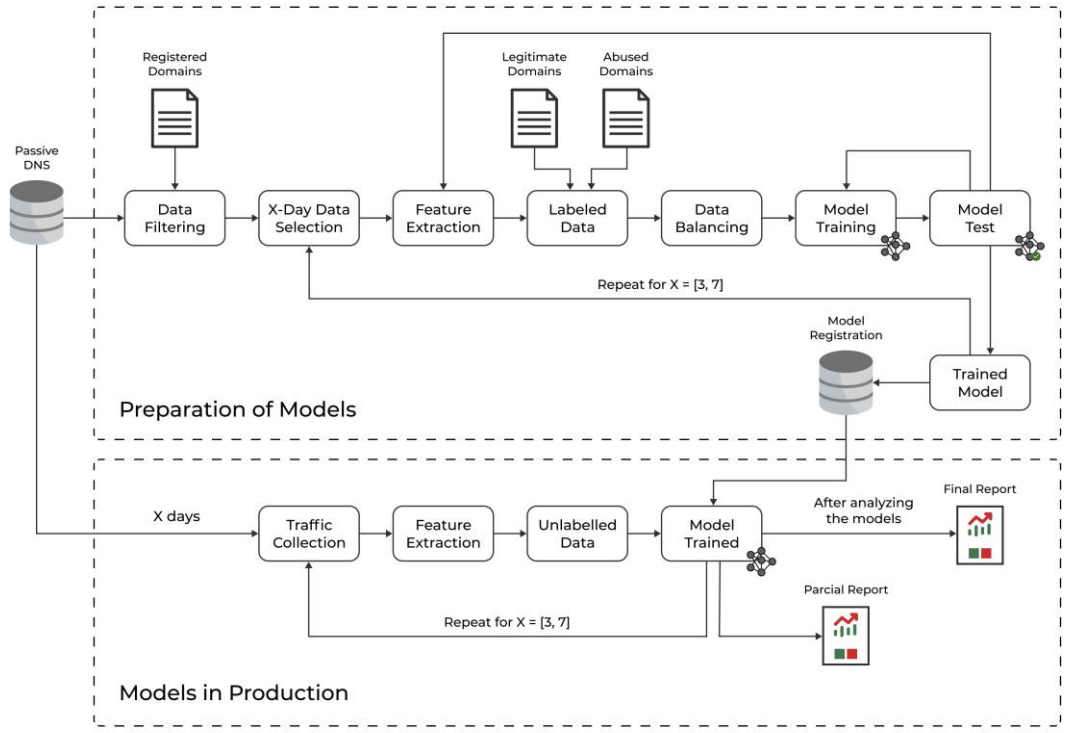

Figure 1. System overview

Table 1. Features extracted from passive DNS and their definitions

Feature	Definition
nb_days_until_collect*	No. of days from domain registration to first query.
nb_days*	No. of days collected.
nb_domain_queries	No. of queries for the domain.
nb_qnames	Count of distinct Query Names (QNAMEs).
min_ttl	Minimum TTL.
ttl_changes	TTL changes count.
avg_prot*	Average of the protocol column – TCP (6) or UDP (17).
nb_ips	Count of distinct IP addresses.
frequent_aa	Most frequent Authoritative Answer (AA).
frequent_cd*	Most frequent Checking Disabled (CD) – Domain Name System Security Extensions (DNSSEC).
avg_ancount	Average of the Answer Count (ANCOUNT) column.
avg_arcount	Average of the Additional Information Count (ARCOUNT) column.
avg_nscount	Average of the Authority Count (NSCOUNT) column.
frequent_rcode	Most frequent Response Code (RCODE).
avg_qtype	Average of the Query Type (QTYPE) column.
nb_countries	Count of different countries.
frequent_country	Most frequent country.
nb_asns	Count of distinct ASNs.
avg_labels*	Average of the QNAME labels column.
avg_res_len	Average length of the DNS response message.

3.4 Labeled Data

The blocklist used in data labeling was also made available by the TLD team. Then it was necessary to filter only abused domains in the period of the registered domains list, which contained 4,815 domains identified as abused. Therefore, all abused domains were labeled as 1 (one), and the rest were labeled as 0 (zero), consisting of the legitimate domains. The system only works with newly registered domains, there is no proper list of recently registered legitimate domains, so the previous approach was applied. From the data labeling, the proportions of the domain classes generated corresponding to the periods of three and seven days were identical, 88,926 legitimate and 1,062 abused.

3.5 Data Balancing

As noted in the data labeling, there is more data from legitimate than abused domains, where abused domains only account for 1.18% of the dataset. Thus, it is needed to balance the data by applying the combination of undersampling and oversampling techniques. The data from the legitimate domains are equivalent to the majority class, and in contrast, the data from the abused domains are from the minority class. Therefore, two undersampling techniques, RUS and CC, and five oversampling techniques, ROS, SMOTE, Borderline-SMOTE, SVM-SMOTE, and K-Means SMOTE, are compared in this work. The undersampling and oversampling techniques with the best results in Area Under the ROC Curve (AUC) are combined.

Based on data balancing techniques, it is expected to reduce the majority class data to twice the minority class, which results in 2,124 legitimate domains, and then the minority class is doubled to match the majority class. The end of applying the techniques results in 2,124 legitimate domains and 2,124 abused domains. It is important to emphasize that the number of abused domains only doubles due to the overfitting risk as more synthetic data are created, which is when the model is suitable only for training data. Care with the data balancing rate and model validation are essential at this stage.

3.6 Model Training and Test

When the data is balanced, there is the training step of each model with tree-based ML algorithms, being DT, RF, Gradient Boosting Machine (GBM), XGBoost, and LightGBM, where the algorithm that results in the best AUC in training and test is applied on the system. As tree-based algorithms are used, there is no need to normalize the data, as these algorithms do not suffer from denormalized data [26]. Hyperparameters in ML algorithms are variables determined before the training process, which controls the entire learning stage. Thus, the hyperparameters used were the standards of the scikit-learn library in DT, RF, and GBM, and in XGBoost and LightGBM, there are their libraries. Hyperparameters tuning is required to obtain maximum performance from the XGBoost and LightGBM algorithms and prevent overfitting. One way of fitting is Bayesian optimization, which selects the best hyperparameters by locating the global minimum of the function in the smallest number of possible iterations [7]. When performing Bayesian optimization, it is necessary to inform the number of initial points and iterations, where 50 initial points and 500 iterations were defined.

Regarding the optimized hyperparameters, in XGBoost they are eta, num. estimators, max. depth, subsample, gamma e reg. lambda was optimized. In LightGBM, the hyperparameters were lambda L1 e L2; num. leaves; feature fraction; min. child samples, bagging freq. e bagging fraction. A little about the algorithms based on the boosting method, XGBoost has high scalability in all scenarios, making it possible to run from a simple desktop to distributed machines [27]. The LightGBM was created to accelerate the learning process further, relying on optimizations during the construction of trees [28].

In training and testing the models, the stratified K-fold cross-validation technique with $K = 5$ is used, in which the objective is to identify overfitting and underfitting in the models. The essential idea of the technique is that each dataset sample is tested. At each iteration, the dataset is divided into K folds, with the training being performed in $K - 1$ folds and the test in the part not used in training. In stratified cross-validation, the proportion present in the dataset is maintained in each division of the folds, that is, the balance of the data is conserved in each of the generated folds [29]. Regarding the choice of the value of K, it was defined due to the size of the dataset and is an interesting value for analyzing the models without a high computational cost [30].

Assessment metrics collected during model training and testing are accuracy, precision, recall, F1-score, AUC, and training time. However, the ones considered in comparing the algorithms are AUC and training time. The AUC is the calculation of the area under the Receiver Operating Characteristic (ROC) curve, being a useful metric to evaluate models and increasingly used in ML communities [31], besides there is an equal concern between the classes, that is, the data must be balanced. Training time is applied to evaluate the algorithms in terms of speed. XGBoost and LightGBM are expected to have the shortest time, especially because of parallel computing.

3.7 Models in Operation

After constructing the three and seven-day models, the DNS traffic collection for a specific domain starts for a period of X days, where the value of X depends on the model in use.

Afterward, the features are extracted, and the data is not labeled, and then the domain is classified in one of the models, and a partial classification report is generated. This process in question is repeated until the domain is classified in both models. In the end, there is the final classification report of this domain, where it is shown which class the domain belongs to and the percentage of being legitimate or abused. With this, the system serves as an aid in the early identification of abused domains, especially domains that bypass manual detection methods in TLDs.

4. Results and Discussion

In this section, the results obtained in this work and the discussion will be compared to other works dealing with the same theme. Thus, the results of comparison of undersampling and oversampling techniques, algorithm performance, importance of each feature, and validation of the model in a test environment. A machine with the following technical specifications was used to obtain the results: Intel Xeon E52650 processor with 2.30GHz (10 cores and 20 threads); 32GB DDR4 RAM; 300GB storage (SSD); and Ubuntu Server 18.04 LTS operating system.

4.1 Comparison of Resampling Techniques

From the combination of undersampling and oversampling techniques in the three-day data and training with LightGBM with Bayesian optimization, Table 2 was built to compare techniques in relation to the AUC metric. In the comparison of undersampling techniques, it is noted that CC was the best, particularly because of its functioning, being able to obtain data from legitimate domains in different regions of space. Moreover, when combining the techniques of undersampling and oversampling, there is a considerable increase in AUC.

Table 2. Comparison of data balancing techniques

Data Balancing Technique		AUC
Undersampling	**Oversampling**	
RUS	Without Oversampling	0.8602
	ROS	0.9229
	SMOTE	0.9221
	Borderline SMOTE	0.9182
	SVM SMOTE	0.9214
	K-Means SMOTE	0.9341
CC	Without Oversampling	0.9342
	ROS	0.9663
	SMOTE	0.9657
	Borderline SMOTE	0.9668
	SVM SMOTE	0.9662
	K-Means SMOTE	0.9673

As far as oversampling techniques are concerned, all the techniques generally showed excellent results, with minimal variation. However, the technique that showed the highest AUC was the K-Means SMOTE, being the newest found in the literature and having a different operation from other techniques. Therefore, the combination of CC and K-Means SMOTE techniques is used to balance the data.

Comparing this work with other works found that apply passive DNS, as presented above, only in Bao *et al.* [22], Wang *et al.* [23], and Silveira *et al.* [7] are discussed the issue of data balancing, which is done through downsampling, K-Means SMOTE, and RUS, respectively. However, in none of the works is the use of several techniques compared, besides the use of CC and the combination of undersampling and oversampling.

4.2 Algorithms Performance

The results obtained in the training and testing of the DT, RF, GBM, XGBoost and LightGBM algorithms in the three-day model are shown in Table 3. As mentioned in Section III, the metrics analyzed to compare algorithms are AUC and training time (in seconds). When the algorithm is followed by (BO), it indicates that Bayesian optimization was applied to select the hyperparameters. Finally, it is essential to highlight that the AUC refers to the average AUC. In each training and test, its respective ROC curve is built with a certain area. Thus, as the value of K is equal to 5, then five ROC curves are generated with their respective AUC, and at the end, the average AUC is calculated.

Table 3. Algorithm's performance in training and testing the three-day model

Algorithm	Metrics	
	AUC	**Time (s)**
DT	0.8765	0.129712
RF	0.9582	2.091682
GBM	0.9600	1.916872
XGBoost	0.9642	0.904666
LightGBM	0.9669	1.583613
XGBoost (BO)	0.9663	4.865108
LightGBM (BO)	0.9673	1.898552

From the results in Table 3 in terms of AUC, it can be noted that the algorithms based on the boosting method had the best values, in which the AUC was bigger than or equal to 0.9600. However, the algorithm with the highest AUC consists of LightGBM using Bayesian optimization, where the average AUC is 0.9673. The DT algorithm obtained the shortest training time but reached the lowest AUC in the comparison. Thus, comparing only the time in algorithms based on the boosting method, GBM took the longest, and XGBoost without Bayesian optimization was the fastest. When considering the adjustment of the hyperparameters, the scenario changes, and LightGBM presents the best training time and the highest AUC, so it is chosen for training the model. It is noteworthy that LightGBM and XGBoost do not present so much time difference because they have little training data and the hardware used. However, due to parallel computing, the speed in the training stage is a positive point to apply these algorithms in constructing an ML model. To further detail the results achieved in terms of AUC, in addition to presenting the ROC curve in each algorithm, Figure 2 shows the ROC and AUC curves with the training of (a) DT, (b) RF, (c) GBM, (d) XGBoost, (e) LightGBM, (f) XGBoost (BO) and (g) LightGBM (BO) in the three-day model. In the legend of each graph, there is the AUC in each of the folds, ranging from fold 0 to fold 4. In addition, there is a variation between each ROC curve after displaying the average AUC, which is indicated in parentheses. The graphs showed a low variation in the ROC curves, which is equal to 0.01.

Table 4 shows the metrics achieved in the training and testing of the algorithms in the seven-day model. Therefore, as in the three-day model, the LightGBM (BO) had the best average AUC of 0.9674, in addition to a training time of only 2.152275 seconds. With that, it is possible to verify again the excellent results obtained in LightGBM, and thus, its applicability for the construction of the model. Results can improve further with increasing initial points and iterations in Bayesian optimization, but it depends on the hardware and how much is feasible for the situation.

Figure 3 shows the ROC and AUC curves in the seven-day

model achieved in the training of (a) DT, (b) RF, (c) GBM, (d) XGBoost, (e) LightGBM, (f) XGBoost (BO) and (g) LightGBM (BO). As can be seen, the maximum variation in the ROC curves was 0.02, and with that, there is a very low variation in general. Consequently, the AUC in each fold is similar to each other.

Table 4. Algorithm's performance in training and testing the seven-day model

Algorithm	Metrics	
	AUC	Time (s)
DT	0.8748	0.137400
RF	0.9548	1.896597
GBM	0.9569	2.080640
XGBoost	0.9633	0.921984
LightGBM	0.9661	1.700336
XGBoost (BO)	0.9647	4.073478
LightGBM (BO)	0.9674	2.152275

The results obtained found that the average AUC is 0.9673 and 0.9674 for the three-day and seven-day models, respectively. When averaging, the system obtains an overall AUC of 0.96735. Compared with other works that follow the same line, Exposure [20] obtained an AUC of 0.987 using DT and cross-validation with $K = 10$. In the work by Lison *et al*. [21], neural networks with 12 tests were used, which had the lowest AUC of 0.976 and the highest AUC of 0.997. In Bao *et al*. [22], an AUC of 0.994 was achieved from the full use of features, in addition to using XGBoost. Finally, in Silveira *et al*. [7], XGBoost is also applied and achieved an AUC of 0.976. The works use lexical and numerical features to obtain these results [20,21,22], except for certain works [19,9,7]. Regarding the ML algorithm, none of the works uses LightGBM, using at most XGBoost [22,7] and CatBoost [23]. In summary, this work uses exclusively the combination of CC and K-Means SMOTE techniques to balance the data. In addition, LightGBM is employed to train the models and the use of exclusive passive DNS features in conjunction with columns enriched in ENTRADA, where DNS traffic is collected from an authoritative TLD server. Based on this, the work presented an overall AUC of 0.96735, corresponding to the works shown, with the addition that consists in the early identification of an abused domain in TLD, being able to monitor the newly registered domains for three and seven days after the first DNS query.

4.3 Importance of Each Feature

With the generation of the three-day model, the importance of the features for the classification of a newly registered domain was extracted, where the feature importance graph is shown in Figure 4. The three features that are pointed out as the most important are: the "nb_days_until_collect", since the abused domains tend to be registered and soon after used, unlike the legitimate domains; the "min_ttl", which abused domains use low TTL values to have more IP changes, especially for fast-flux domains attacks, besides the direct influence on the "ttl_changes" feature [20]; and finally, "avg_res_len", where the variation in the size of the DNS response between legitimate and abused domains can be noted.

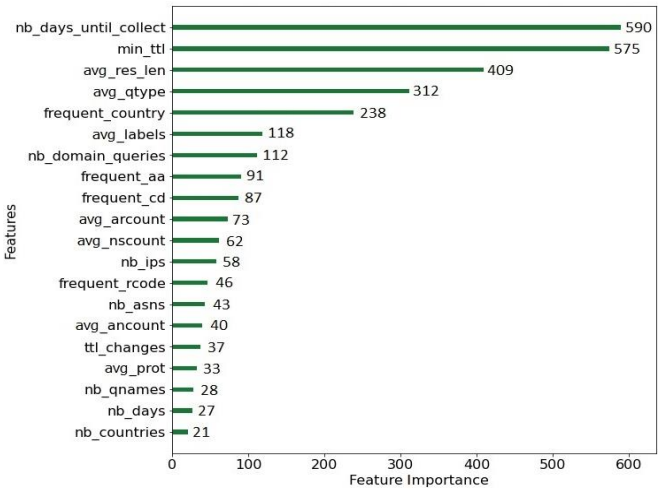

Figure 4. Importance of each feature in the three-day model

About the importance of other extracted features, the "nb_domain_queries" is used because of the abused domains for phishing, which generally have a high number of queries in a short period [21]. The features "nb_countries", "nb_asns", and "frequent_country" refer to location since the abused domains resolve to affected machines in different places in the world. Therefore, in the abused domains, different ASNs are defined to vary in the IP prefixes [9]. Finally, as DNS traffic is collected from an authoritative TLD server, this work compares with Kopis [9].

4.4 Models Validation in Test Environment

After generating the three and seven-day models, the models were validated in a test environment. The newly registered domains were tested corresponding to the beginning of January until the end of June 2021, highlighting that the collection of the DNS traffic is until early September. These domains were classified in the models, and the confusion matrices generated from the classification of three and seven days are presented in Tables 5 and 6, respectively.

Based on each confusion matrix and calculating the TPR and FPR, in the three-day model, there was a TPR of 0.8656, corresponding to how many of the abused domains were correctly classified, and FPR of 0.3471, which designates the number of abused domains misclassified. In turn, the seven-day model had a TPR of 0.8682 and an FPR of 0.3216. In general, the system presented an interesting result in the test environment, especially when the objective was to detect the abused domains. However, the models did not do very well for the legitimate domains, which generated many FPs, probably due to oversampling in the data.

Table 5. Model validation confusion matrix (three-day)

	Predicted Label		
		Abused	Legitimate
True Label	Abused	335	52
	Legitimate	9,917	18,657

Table 6. Model validation confusion matrix (seven-day)

	Predicted Label		
		Abused	Legitimate
True Label	Abused	336	51
	Legitimate	9,189	19,385

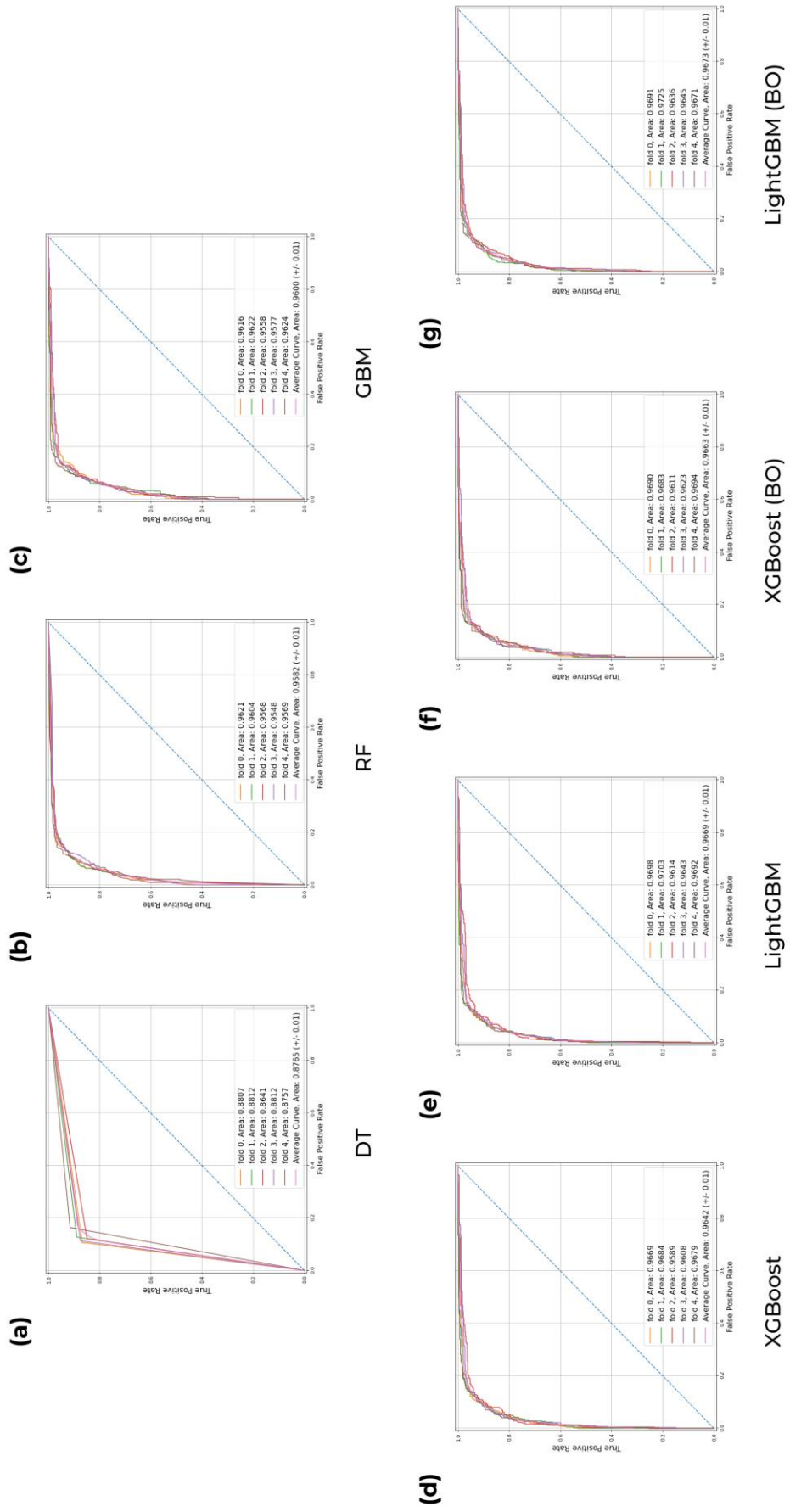

Figure 2: AUC of each fold using the (a) DT, (b) RF, (c) GBM, (d) XGBoost, (e) LightGBM, (f) XGBoost (BO) and (g) LightGBM (BO) in the three-day model

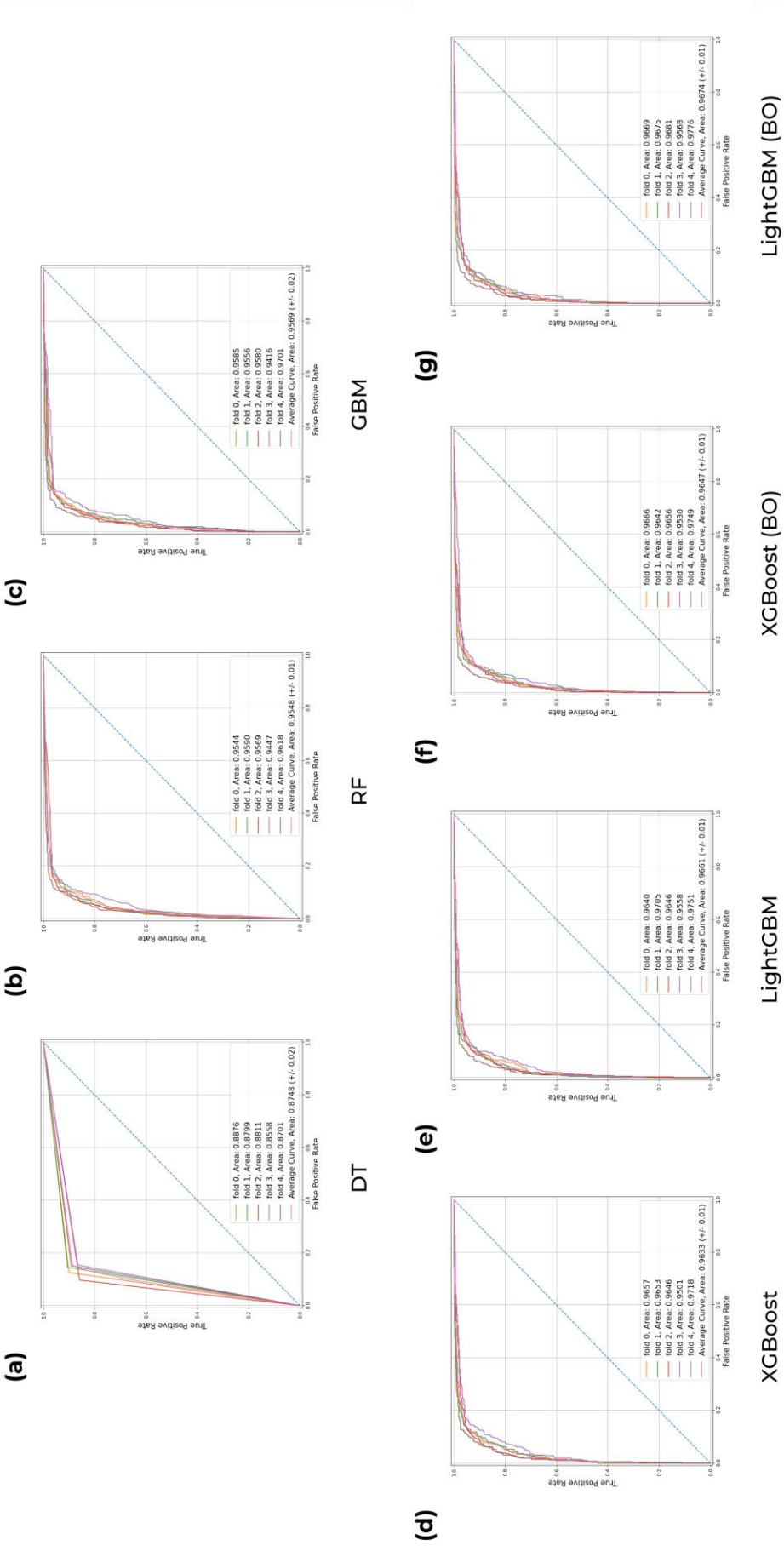

Figure 3: AUC of each fold using the (a) DT, (b) RF, (c) GBM, (d) XGBoost, (e) LightGBM, (f) XGBoost (BO) and (g) LightGBM (BO) in the seven-day model

5. Conclusions

In this work, a system capable of performing the early identification of abused domains in TLD through passive DNS was presented, collected from an authoritative TLD server for 12 months. In the system, 20 features are used, extracted exclusively from the passive DNS and the columns generated from the data enrichment. Thus, the models monitor newly registered domains in their first three and seven days after the first DNS query, in which the behavior of these domains are verified twice. Because the data are unbalanced, CC and K-Means SMOTE techniques were applied. For the training of the models, the LightGBM algorithm was used, applying Bayesian optimization with 50 initial points and 500 iterations to select the best hyperparameters. When evaluating the models in the training and testing stage, the three-day model had an average AUC of 0.9673, and the seven-day model achieved an average AUC of 0.9674, in addition to the low training time with an average of 2 seconds. Three and seven-day models were used in a test environment, obtaining a TPR of 0.8656 and 0.8682, respectively. Therefore, it is observed that the models had good results in identifying abused domains in the tests. Finally, the importance of a TLD taking advantage of the ability to identify newly abused domains quickly is highlighted, and from there, mitigate these domains, preventing users from falling into scams or companies from suffering losses [32].

Regarding future works, there is an increase in data to reduce the amount of synthetic data generated. With that, the models probably have a better performance in the production environment, and the implementation of incremental training to the models seeking to combat the misclassifications, in which knowledge is added over time. As a way to help identify more abused domains, unsupervised learning can be used. Finally, it emphasizes the importance of monitoring newly registered domains and building models to monitor domains in the second, third and fourth week, thus analyzing their first month of life.

6. Acknowledgement

The authors thanks Brazilian Network Information Center (NIC.br) for funding this research, under the Foundation for the Development of UNESP (Fundunesp) Process, number 2764/2018.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES).

References

[1] D. N. Stat, “Domain name registration’s statistics,” 2022, URL: https://domainnamestat.com/statistics/overview, [Online; accessed on January 23, 2022].

[2] S. Khalid, A. Mahboob, F. Azim, A. U. Rehman, “IDHOCNET-A novel protocol stack and architecture for ad hoc networks,” International Journal of Communication Networks and Information Security (IJCNIS), Vol. 7, No. 1, pp. 20, 2015.

[3] K. R. Fall, W. R. Stevens, “TCP/IP illustrated, volume 1: the protocols,” Addison-Wesley, 2011.

[4] J. F. Kurose, K. W. Ross, “Computer Networking: A Top-Down Approach,” Pearson, 2017.

[5] L. Desmet, J. Spooren, T. Vissers, P. Janssen, W. Joosen, “Premadoma: an operational solution to prevent malicious domain name registrations in the .eu TLD,” Digital Threats: Research and Practice, Vol. 2, No. 1, pp. 1-24, 2021.

[6] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, R. Joffe, “Enabling network security through active DNS datasets,” International Symposium on Research in Attacks, Intrusions, and Defenses, pp. 188-208, 2016.

[7] M. R. Silveira, L. M. Da Silva, A. M. Cansian, H. K. Kobayashi, “XGBoost applied to identify malicious domains using passive DNS,” 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), pp. 1-4, 2020.

[8] F. Weimer, “Passive DNS replication,” FIRST Conference on Computer Security Incident, pp. 1-14, 2005.

[9] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, D. Dagon, “Detecting malware domains at the upper DNS hierarchy,” Proceedings of the 20th USENIX Security Symposium, Vol. 11, pp. 1-16, 2011.

[10] T. Kulikova, T. Shcherbakova, “Spam and Phishing in Q3 2021,” 2021, URL: https://securelist.com/spam-and-phishing-in-q3-2021/104741/, [Online; accessed on December 13, 2021].

[11] Symantec, “Internet security threat report,” Vol. 21, 2019, URL: https://docs.broadcom.com/doc/istr-24-2019-en, [Online; accessed on August 17, 2021].

[12] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, F. Herrera, “Learning from imbalanced data sets,” Springer, 2018.

[13] S. J. Yen, Y. S. Lee, “Cluster-based under-sampling approaches for imbalanced data distributions,” Expert Systems with Applications, Vol. 36, No. 3, pp. 5718-5727, 2009.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” Journal of Artificial Intelligence Research, Vol. 16, pp. 321-357, 2002.

[15] H. Han, W. Y. Wang, B. H. Mao, “Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning,” International Conference on Intelligent Computing, pp. 878-887, 2005.

[16] H. M. Nguyen, E. W. Cooper, K. Kamei, “Borderline over-sampling for imbalanced data classification,” International Journal of Knowledge Engineering and Soft Data Paradigms, Vol. 3, No. 1, pp. 4-21, 2011.

[17] G. Douzas, F. Bacao, F. Last, “Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE,” Information Sciences, Vol. 465, pp. 1-20, 2018.

[18] M. Wullink, G. C. M. Moura, M. Müller, C. Hesselman, “ENTRADA: A high-performance network traffic data streaming warehouse,” NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, pp. 913-918, 2016.

[19] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, N. Feamster, “Building a dynamic reputation system for DNS,” Proceedings of the 19th USENIX Security Symposium, pp. 273-290, 2010.

[20] L. Bilge, E. Kirda, C. Kruegel, M. Balduzzi, “EXPOSURE: finding malicious domains using passive DNS analysis,” Ndss, pp. 1-17, 2011.

[21] P. Lison , V. Mavroeidis, “Neural reputation models learned from passive DNS data,” 2017 IEEE International Conference on Big Data (Big Data), pp. 3662-3671, 2017.

[22] Z. Bao, W. Wang, Y. Lan, “Using passive DNS to detect malicious domain name,” Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, pp. 1-8, 2019.

[23] Q. Wang, L. Li, B. Jiang, Z. Lu, J. Liu, S. Jian, “Malicious domain detection based on k-means and SMOTE,” International Conference on Computational Science, pp. 468-481, 2020.

[24] L. Watkins, S. Beck, J. Zook, A. Buczak, J. Chavis, W. H. Robinson, J. A. Morales, S. Mishra, “Using semi-supervised machine learning to address the big data problem in DNS networks,” 2017 IEEE 7th Annual Computing and

Communication Workshop and Conference (CCWC), pp. 1-6, 2017.
[25] I. Khalil, T. Yu, B. Guan, "Discovering Malicious Domains through Passive DNS Data Graph Analysis," Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, pp. 663-674, 2016.
[26] D. Borkin, A. Némethová, G. Michaľčonok, K. Maiorov, "Impact of data normalization on classification model accuracy," Research Papers Faculty of Materials Science and Technology Slovak University of Technology, Vol. 27, No. 45, pp. 79-84, 2019.
[27] T. Chen, C. Guestrin, "XGBoost: A scalable tree boosting system," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016.
[28] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," Advances in Neural Information Processing Systems 30 (NIPS 2017), Vol. 30, pp. 3146-3154, 2017.
[29] J. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, "Classification and regression trees," CRC Press, 1984.
[30] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," Appears in the International Joint Conference on Artificial Intelligence (IJCAI), Vol. 14, No. 2, pp. 1137-1145, 1995.
[31] T. Fawcett, "ROC graphs: notes and practical considerations for researchers," Machine Learning, Vol. 31, No. 1, pp. 1-38, 2004.
[32] L. M. Da Silva, M. R. Silveira, A. M. Cansian, H. K. Kobayashi, "Multiclass classification of malicious domains using passive DNS with XGBoost:(work in progress)," 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), pp. 1-3, 2020.