Journal of Computer Networks and Communications

Security Solutions for IoT and Cyber-Physical Systems

Lead Guest Editor: Mojtaba Alizadeh
Guest Editors: Mahdi Zareei and Saman S. Chaeikar

Security Solutions for IoT and Cyber-Physical Systems

Journal of Computer Networks and Communications

Security Solutions for IoT and Cyber-Physical Systems

Lead Guest Editor: Mojtaba Alizadeh
Guest Editors: Mahdi Zareei and Saman S. Chaeikar

Academic Editors

Contents

Hindawi
Journal of Computer Networks and Communications
Volume 2021, Article ID 4767388, 13 pages
https://doi.org/10.1155/2021/4767388

Research Article

A Novel Approach for Detecting DGA-Based Botnets in DNS Queries Using Machine Learning Techniques

Ali Soleymani and Fatemeh Arabgol

Faculty of Computer Engineering, Iranians University an e-Institute of Higher Education, Tehran, Iran

Correspondence should be addressed to Ali Soleymani; ali.soleymani@iranian.ac.ir

Received 26 April 2021; Accepted 24 June 2021; Published 5 July 2021

Academic Editor: Saman Shojae Chaeikar

In today's security landscape, advanced threats are becoming increasingly difficult to detect as the pattern of attacks expands. Classical approaches that rely heavily on static matching, such as blacklisting or regular expression patterns, may be limited in flexibility or uncertainty in detecting malicious data in system data. This is where machine learning techniques can show their value and provide new insights and higher detection rates. The behavior of botnets that use domain-flux techniques to hide command and control channels was investigated in this research. The machine learning algorithm and text mining used to analyze the network DNS protocol and identify botnets were also described. For this purpose, extracted and labeled domain name datasets containing healthy and infected DGA botnet data were used. Data preprocessing techniques based on a text-mining approach were applied to explore domain name strings with n-gram analysis and PCA. Its performance is improved by extracting statistical features by principal component analysis. The performance of the proposed model has been evaluated using different classifiers of machine learning algorithms such as decision tree, support vector machine, random forest, and logistic regression. Experimental results show that the random forest algorithm can be used effectively in botnet detection and has the best botnet detection accuracy.

1. Introduction

The popularity of using the Internet has led to some dangers of network attacks, including botnets, DDoS attacks, and spam. Nowadays, botnets are the most widespread and serious threat that commonly occurs in cyberattacks. Bots are controlled by an attacker, called a botmaster, under a shared command and control (C&C) infrastructure that allows them to control infected computer systems remotely. Bots are different from other forms of malware in that they are highly autonomous and are equipped with the ability to use communication channels to receive commands and code updates from their control system. They can also notify their working status to their control system periodically. The botnet control system, or command and control servers, is the means by which the botmaster sends commands and code updates to bots. Botnets are often used to transmit malware, send spams, steal sensitive information, deceive, generate virtual clicks, or more seriously carry out large-scale network attacks, such as DDoS attacks. According to some security reports, about 80% of Internet traffic is related to the activities of botnets, including spamming and network attacks [1]. Domain Name Service (DNS) is an essential service on the Internet that allows the resolution of hostnames, or domain names to Internet Protocol (IP) addresses and vice versa. For example, every time a web client browser accesses a web page, it first sends a request to the DNS system to find the IP address of the web server. Next, it uses the IP address to access the web server and load the requested web page. Most of the legitimate applications use DNS services when making requests for accessing network services. However, DNS services are also used by bots of botnets as legitimate applications. Bots send DNS queries to find the IP address of the C&C server and when they have an IP address; they access the C&C server to receive commands, as well as to download the updated bot code. To evade the scanning and detecting C&C servers, the botmaster is constantly changing the names and IP addresses of C&C servers using predefined techniques, such as domain generation algorithms (DGA), or Fast Flux [2, 3]. The names and IP addresses of C&C servers are constantly being pushed to the DNS system. Bots are also able to automatically generate C&C server names in accordance with these techniques. As a result, bots can still find the IP addresses of C&C servers by generating their hostnames automatically and using these hostnames to query the DNS service. Therefore, monitoring and analyzing DNS query data can reveal the existence of malicious activities in the monitored network, since some of the DNS query data may be generated by botnets. An in-depth analysis of suspicious DNS queries may reveal valuable information regarding the presence of C&C servers and botnets. Dealing effectively with botnets requires careful consideration of the channels of control to control over them. This has become one of the major challenges for security systems around the world.

Botnets are widely used in organized crime to infiltrate the security systems of governments, banks, and corporations. In recent years, many research studies has been done on methods of detecting and preventing botnets. According to Ryu and Yang [4], the number of IP addresses created as a control and command server hosted by Amazon in 2017 has increased 6 times against that of 2016 [4]. Attackers use various methods such as encryption and new communication protocols to strengthen the basis for sending their commands. Based on [5], the primary goals of botnets are as follows: (1) information dispersion: sending spams, distributed, denying service, distributing false information from illegal sources, and eliminating or reducing bandwidth; (2) information harvesting: obtaining personal identities, passwords, and financial information; and (3) information processing: information processing to crack the password to access other hosts.

In 1993, when the EGGDROP botnet was reported as the first botnet, the Necurs botnet was one of the most active malware publishers in 2016. More than 2.3 million spams, including JavaScript and Visual Basic downloaders, were sent as e-mail attachments per day on November 24 by botnet Necurs. According to the same report, botnet showed the largest DDoS attack in 2016, recorded by the French host company OVH. It was with a maximum speed of 1 TBps, targeting Internet of Things (IoT) devices such as home routers and IP cameras, as Gartner predicted. There will be more than 12 billion IoT devices by 2022, so the first step in preventing these threats is to detect them, which has been the subject of much research in recent years [4].

A DGA detection framework, called Deep Bot Detect (DBD) proposed to analyze and categorize statistical features of DNS queries extracted by machine learning. The results of the proposed framework prove the accuracy and low false-positive rate to detect domain-flux botnets [6].

Bilbo is a hybrid model which is the composition of convolutional neural network (CNN), long short-term memory (LSTM), and artificial neural network (ANN) proposed to detect DGA botnets [7]. The experiments are performed on three DGA dictionaries: gozi, matsnu, and suppobox. From the datasets, 80% was used for training while 20% was randomly selected for testing and holdout. For the results of testing classification, generalizability, and time-based resiliency, Bilbo successfully classified traffic matching the expected network pattern. Although the identified domains from the network logs were not botnets or worms reaching out to a C&C, which are very rare, Bilbo was able to identify dictionary DGAs used by advertisement networks and other applications with potentially malicious intent. Deployment on real world: the performance of the system was evaluated using data from Alexa top 1 million DGArchive.

There are various challenges in botnet detection algorithms which may affect the results [8]. These challenges are related to the quality and quantity of the datasets for training and testing the methods based on machine learning. Another challenge is the Fast Flux method in which a botnet can hide its identity and cybercriminals can evade or detect it. While Deep Packet Inspection (DPI) is not effective on encrypted traffic, machine learning botnet detection and host-based detection mechanism require many resources like processing and storage. This may cause overhead in the hosts because they must keep running to inspect network traffic and collect data.

Flow-based functions such as source and destination IP, protocol, and number of packets sent or received are the most commonly used functions in the field of bot inspection. However, these functions cannot fully capture communication patterns that may expose other aspects of malicious hosts. Furthermore, the flow-level model generates a high computational overhead, which can be avoided by adjusting the behavioral characteristics such as modifying the structure of the data packet. To overcome these limitations, exploring the graph-based features of these methods is the most promising way for future research, where the graph is drawn from the host-to-host network flow communication pattern [9, 10].

A method presented based on analyzing the similar periodic time intervals series of DNS queries to identify DGA-bot-infected machines [11]. To measure the similar periodicity of DNS queries, the squared Euclidean distance between each pair of their time interval series is calculated. Finally, they apply a hierarchical clustering algorithm to cluster high similar domain names. The results show that the domain names are generated by the same botnet or DGA would be grouped into the same cluster.

2. Methodology

As mentioned before, the main feature of botnets is the C&C infrastructure for them. Each botnet is a coordinated group of bots that are routed through C&C channels and perform malicious activities. Thus, the main purpose of the proposed method is to detect the botnet to prevent the spread of spam and network traffic. Millions of spams can be prevented if botnets are discovered.

In the proposed method, the user behavior is analyzed and the amount of traffic transferred between hosts is recorded to extract the network behavior patterns. Term frequency-inverse document frequency (TFIDF) models are used to model a module for detecting behavioral patterns, and principal component analysis (PCA) is used to increase the speed and accuracy of diagnosis evaluation results. Behavioral patterns can be extracted from a set of attack packages and used in the form of intelligible detection rules to detect online intrusions. Naturally, the more the varied and high-quality data is used, the further the strength of the proposed method is evaluated. In this way, processes are implemented on how to use machine learning to detect malicious DGA domains. This helps to extend existing security applications to Splunk.

Splunk is a software platform to search, analyze, and visualize the machine-generated data gathered from websites, applications, sensors, and devices. To implement the proposed method, Splunk version 7.0 is used on a PC with 16 GB of memory, Intel Core i7-7660U processor, and 64-bit Windows 10 operating system. Following plugins are used to identify malicious domain names:

(i) *DGA Analysis App*. This app shows how to operationalize machine learning using MLTK to detect malicious domain names. Malware like botnets uses domain generation algorithms to create URLs that host malicious websites or C&C servers. Static matching does not always help. Therefore, machine learning models can add value and allow increasing detection rates [12].

(ii) *Splunk Machine Learning Toolkit*. The Splunk Machine Learning Toolkit App delivers new SPL commands, custom visualizations, assistants, and examples to explore a variety of ML concepts. It is also including the ability to apply the visualizations and SPL commands to your data. You can inspect the assistant panels and underlying code to see how it all works [13].

(iii) *Python for Scientific Computing*. This add-on contains a Python interpreter bundled with the following scientific and machine learning libraries: numpy, scipy, pandas, scikit-learn, and stats models. With this add-on, it is possible to import these powerful libraries in a custom search command, custom rest endpoints, modular inputs, and so forth [14].

(iv) *Parallel Coordinates*. Custom visualizations give a new interactive way to visualize data during search and investigation. It offers better communication results in dashboards and reports. After installing this app, a parallel coordinates visualization has been founded as an additional item in the visualization picker in search and dashboard [15].

(v) *3D Scatterplot*. This visualization allows viewing a scatterplot in three dimensions [16].

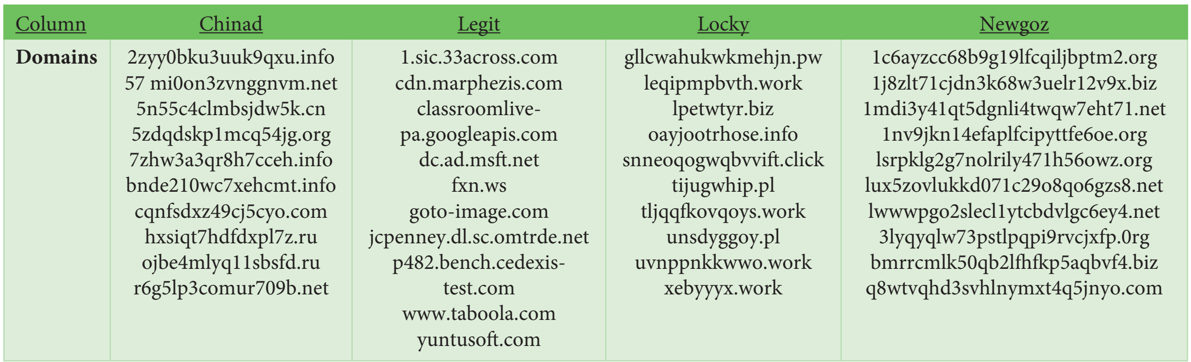
3. Research Data

To evaluate the performance of domain name classification using machine learning algorithms, an extracted and tagged domain name dataset, which has 100,000 domain names, was applied. This includes a collection of harmless domain names taken from the Spamhaus website [17]. The risk-free domain set is at the top of the Alexa rankings. Safe domain names are checked at virustotal.com to make sure they are safe. Almost 60% of domain names are legitimate domains and the remaining 40% belong to three DGA subcategories that correspond to different types of botnets. A list of locky, Chinad, and NewgoZeus botnets and a collection of dangerous domain names are presented in Figure 1. The first dataset starts with tagged domain names that represent a legitimate domain or were created by a DGA [18]. In Figure 2, legitimate domain letters are in blue clusters and dangerous domain letters are in separate clusters (yellow, red, and purple). This indicates that some DGA subcategories such as newGOZ (yellow) are more separable than others (red and purple). It means that the newGOZ detection is more accurate than the locker, Game Over Zeus, and ChinAd detectors. The results of this approach are the first numeric properties that are calculated from the domain name string. Using these results, it is possible to determine how the domain is related to the specifications of the DGA subcategories.

4. Proposed Model

Figure 3 shows the framework model of a botnet detection system based on machine learning using DNS query data. According to this framework, botnets regularly send lookup queries to the DNS system to automatically find the IP addresses of C&C servers using the generated domain names. They are executed in two phases of training and detection. During the training phase, the DNS query data is collected and then the domain names in the DNS queries are extracted. Next, the domain name sets are preprocessed to extract the attributes for training. In the training phase, machine learning algorithms are used to learn the classifiers. After the evaluation process, the machine learning algorithm will be selected to apply in the proposed diagnostic model which has the highest overall classification accuracy. During the detection phase of this model, DNS queries are monitored and go through the process. This process is domain name extraction, preprocessing, and categorization using the generated classification in the training phase, respectively. This is to determine whether the domain name is legitimate or belongs to a botnet. The preprocessing step is the same for all domain letters in the training and the detection phase. However, for all domain names in the training phase dataset, this step is done in offline mode.

Automatically generated botnet domain names usually have domain name specifications and lexical properties that are different from legitimate domain names. Some features such as Shannon entropy criteria, known word rate in a dictionary, domain name, length, consonant, and vowel rate, as well as domain analysis strings, are explored by n-gram analysis and principal component analysis (PCA). Dimensions are effective in improving the analysis of results and increasing the accuracy of diagnosis, using the TFIDF algorithm and setting the n-gram parameter to the character



Column	Chinad	Legit	Locky	Newgoz
Domains	2zyy0bku3uuk9qxu.info 57 mi0on3zvnggnvm.net 5n55c4clmbsjdw5k.cn 5zdqdskp1mcq54jg.org 7zhw3a3qr8h7cceh.info bnde210wc7xehcmt.info cqnfsdxz49cj5cyo.com hxsiqt7hdfdxpl7z.ru ojbe4mlyq11sbsfd.ru r6g5lp3comur709b.net	1.sic.33across.com cdn.marphezis.com classroomlive-pa.googleapis.com dc.ad.msft.net fxn.ws goto-image.com jcpenney.dl.sc.omtrde.net p482.bench.cedexis-test.com www.taboola.com yuntusoft.com	gllcwahukwkmehjn.pw leqipmpbvth.work lpetwtyr.biz oayjootrhose.info snneoqogwqbvvift.click tijugwhip.pl tljqqfkovqoys.work unsdyggoy.pl uvnppnkkwwo.work xebyyyx.work	1c6ayzcc68b9g19lfcqiljbptm2.org 1j8zlt71cjdn3k68w3uelr12v9x.biz 1mdi3y41qt5dgnli4twqw7eht71.net 1nv9jkn14efaplfcipyttfe6oe.org lsrpklg2g7nolrily471h56owz.org lux5zovlukkd071c29o8qo6gzs8.net lwwwpgo2slecl1ytcbdvlgc6ey4.net 3lyqyqlw73pstlpqpi9rvcjxfp.0rg bmrrcmlk50qb2lfhfkp5aqbvf4.biz q8wtvqhd3svhlnymxt4q5jnyo.com

Figure 1: Separation of domains by DGA botnet.

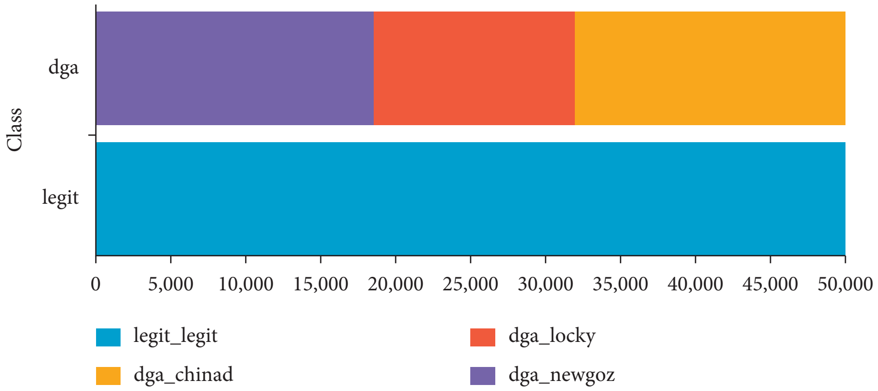


Figure 2: Comparative chart of locky, newgoz, chinad botnets, and legit data.

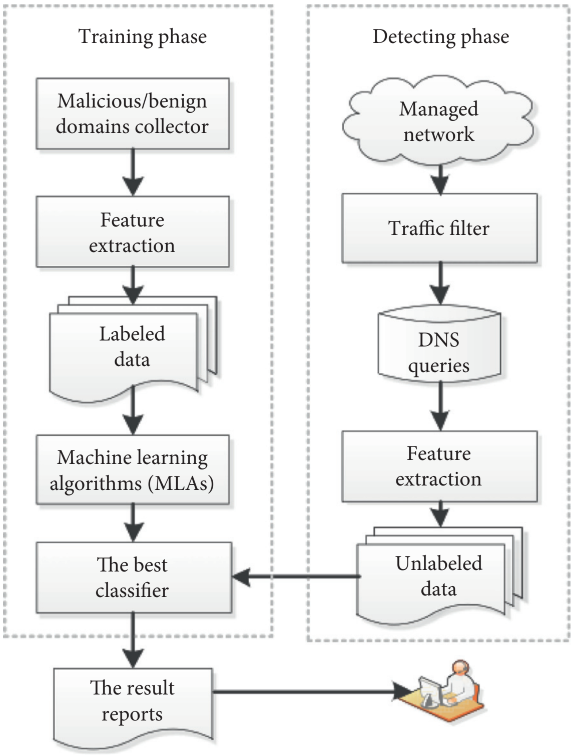


Figure 3: Framework of the detection system. MLA: machine learning algorithm; DNS: domain name server.

length that can be obtained by a matrix. This matrix contains the most frequent groups in the domain name strings. In this way, a high-dimensional result is obtained. Therefore, by using PCA, the dimensions are reduced. PCA provides a useful representation that can be plotted in the form of a three-dimensional point diagram as shown in Figure 4.

Machine learning is applied in Splunk to find anomalies, predict or estimate the response of a system, or cluster to detect behavior. It is often dealt with a set of fields that are unusual information written by a human or the country codes for an IP search, or even a search field from an index.

4.1. Preprocessing Data.

4.1. Preprocessing Data. To identify the features that are most promising for predicting malicious domains, analyze the field SPL command. It is applied to rank all attributes and determine the top-ranking attributes that have the highest accuracy for creating machine learning models. Domain names that are enriched with additional features are shown in Table 1.

The distribution diagram in Figure 5 shows how the two selected features (Shannon entropy and semantic ratio) fit into the legitimate domain and the domain of DGA subcategories. To show the distribution categories and subcategories based on the selected properties, the dependency plot is generated in terms of a parallel coordinate diagram. This allows exploring the relationship between features and identifying patterns in a multidimensional dataset. In this

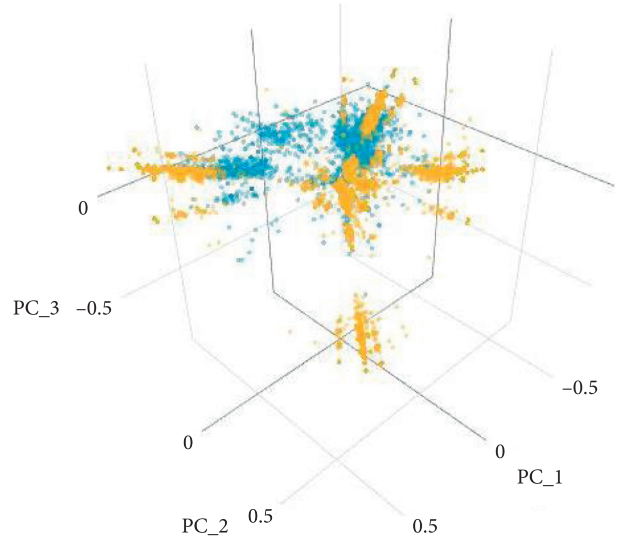


Figure 4: 3D diagram of PCA.

Table 1: Domain name data attributes along with attribute value calculation.

Domain name/attribute	google.com	microsoft.com	g.doubleclick.net	google-analytics.com
Class	legit	legit	legit	legit
partition_number	1	0	0	0
Subclass	legit	legit	legit	legit
ut_consonant_ratio	0.6	0.7	0.7	0.7
ut_digit_ratio	0	0	0	0
ut_domain_length	10	13	17	20
ut_meaning_ratio	0.2	0.692307692	0.823529412	0.2
ut_shannon	2.646439345	3.026986833	3.616874606	3.684184
ut_vowel_ratio	0.4	0.307692308	0.294117647	0.35

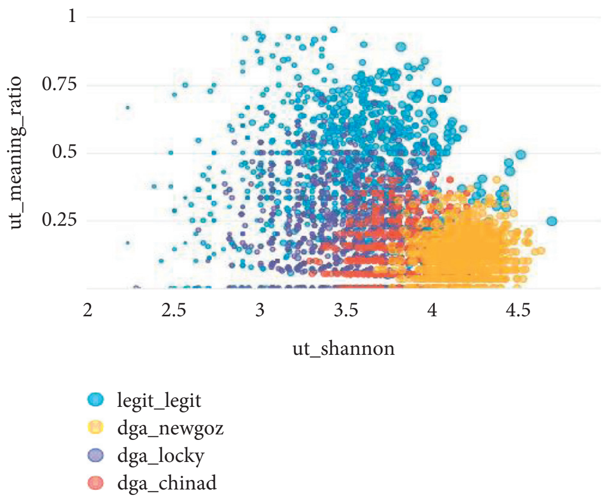


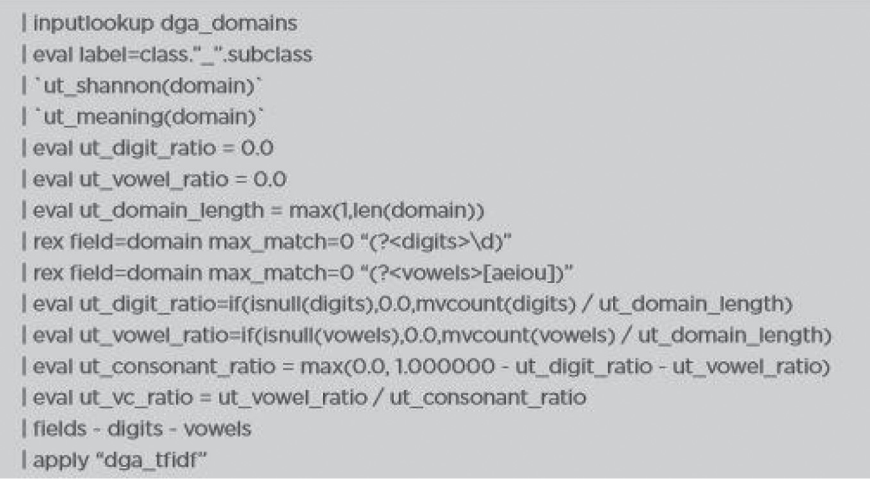
Figure 5: Distribution diagram based on Shannon algorithm and semantic rate.

research, the PCA method is applied to increase the optimization process, which increases the accuracy of the machine learning algorithm. Figure 6 shows the Search Processing Language (SPL) commands for calculating the considered properties.

A dotted graph is presented to identify the combination of properties. Features that represent a distinct distribution can be identified quickly. Therefore, an accurate prediction is more likely to be made. The results of the URL toolbox results can be drawn based on the domain name. The SPL code for calculating values of different domain name properties is shown in Figure 6.

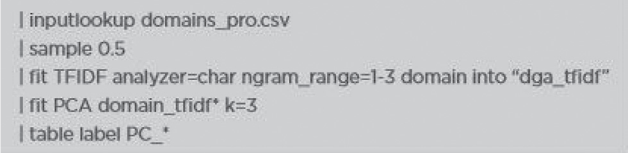
In the feature design and selection phase, certain data can be explored and enriched with additional features that increase the detection accuracy by using machine learning algorithms. Additionally, more features can be obtained by adding, such as Alexa ranking, domain, age, common blacklists, and whitelists. This will result in improving the dataset and identifying features for machine learning. Figure 7 shows the SPL commands for adding PCA features and n-gram analysis and the output of the added fields are demonstrated in Figure 8.

4.2. Classification Criteria. To evaluate the performance of botnet detection techniques, a suitable criterion for quantitative measurement is proposed [19]. In the botnet detection method, the analyzed network data is classified into normal/suspicious groups. Any deviation from the normal traffic pattern is considered suspicious data. Thus, we have to make true positive (TP), true negative (TN), false positive



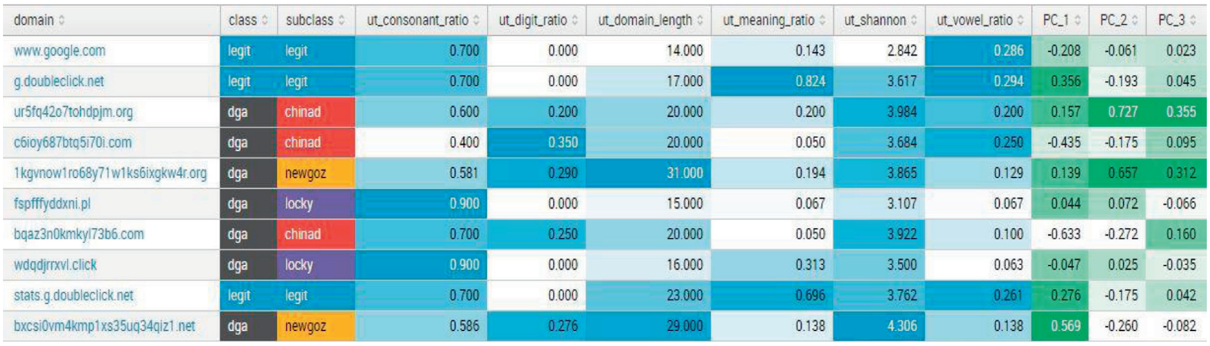
```
| inputlookup dga_domains
| eval label=class."_".subclass
| `ut_shannon(domain)`
| `ut_meaning(domain)`
| eval ut_digit_ratio = 0.0
| eval ut_vowel_ratio = 0.0
| eval ut_domain_length = max(1,len(domain))
| rex field=domain max_match=0 "(?<digits>\d)"
| rex field=domain max_match=0 "(?<vowels>[aeiou])"
| eval ut_digit_ratio=if(isnull(digits),0.0,mvcount(digits) / ut_domain_length)
| eval ut_vowel_ratio=if(isnull(vowels),0.0,mvcount(vowels) / ut_domain_length)
| eval ut_consonant_ratio = max(0.0, 1.000000 - ut_digit_ratio - ut_vowel_ratio)
| eval ut_vc_ratio = ut_vowel_ratio / ut_consonant_ratio
| fields - digits - vowels
| apply "dga_tfidf"
```

Figure 6: Analyze DNS data with SPL language related to feature calculation.



```
| inputlookup domains_pro.csv
| sample 0.5
| fit TFIDF analyzer=char ngram_range=1-3 domain into "dga_tfidf"
| fit PCA domain_tfidf* k=3
| table label PC_*
```

Figure 7: SPL commands in Splank based on the TFIDF model.



domain	class	subclass	ut_consonant_ratio	ut_digit_ratio	ut_domain_length	ut_meaning_ratio	ut_shannon	ut_vowel_ratio	PC_1	PC_2	PC_3
www.google.com	legit	legit	0.700	0.000	14.000	0.143	2.842	0.286	-0.208	-0.061	0.023
g.doubleclick.net	legit	legit	0.700	0.000	17.000	0.824	3.617	0.294	0.356	-0.193	0.045
ur5fq42o7tohdpjm.org	dga	chinad	0.600	0.200	20.000	0.200	3.984	0.200	0.157	0.727	0.355
c6ioy687btq5i70i.com	dga	chinad	0.400	0.350	20.000	0.050	3.684	0.250	-0.435	-0.175	0.095
1kgvnow1ro68y71w1ks6ixgkw4r.org	dga	newgoz	0.581	0.290	31.000	0.194	3.865	0.129	0.139	0.657	0.312
fspfffyddxni.pl	dga	locky	0.900	0.000	15.000	0.067	3.107	0.067	0.044	0.072	-0.066
bqaz3n0kmkyl73b6.com	dga	chinad	0.700	0.250	20.000	0.050	3.922	0.100	-0.633	-0.272	0.160
wdqdjrrxvl.click	dga	locky	0.900	0.000	16.000	0.313	3.500	0.063	-0.047	0.025	-0.035
stats.g.doubleclick.net	legit	legit	0.700	0.000	23.000	0.696	3.762	0.261	0.276	-0.175	0.042
bxcsi0vm4kmp1xs35uq34qiz1.net	dga	newgoz	0.586	0.276	29.000	0.138	4.306	0.138	0.569	-0.260	-0.082

Figure 8: Domain dataset enriched with features.

(FP), and false negative (FN) to determine the true-positive rate (TPR) and the false-positive rate (FPR).

4.2.1. True Positive. The number of correct botnet activity alerts, including attack modes and C&C at each interruption, is called true positive.

4.2.2. True Negative. The number of correct diagnoses of normal activity at each current interruption is called the correct negative.

4.2.3. False Positive. The number of false warnings of botnet activity, including attack and C&C modes at each interruption, is called false positive.

4.2.4. False Negative. The number of false warnings of botnet activity, including attack and C&C modes at each interruption, is called false positive.

4.2.5. Recall. It indicates that the algorithm was able to correctly detect the percentage of bot traffic. This parameter is obtained from the ratio of the number of bot traffic correctly identified as bot traffic by the algorithm to the total number of bot traffic [5] and is calculated by

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{1}$$

4.2.6. Precision. It indicates the percentage of traffic identified by the algorithm as normal traffic is normal. This parameter is obtained from the ratio of the number of normal traffic detected by the algorithm to the total number of traffic detected by the algorithm as normal traffic [5]. It is calculated by

$$\text{precision} = \frac{\text{TN}}{\text{TN} + \text{FN}}. \tag{2}$$

4.2.7. F_Measure. This parameter is a combination of two parameters, precision and recall, and is obtained by

$$F_{\text{measure}} = 2 * \frac{\text{recall} * \text{precision}}{\text{recall} + \text{precision}}. \tag{3}$$

4.2.8. Accuracy. This parameter indicates the overall accuracy of the algorithm and is the ratio of the total number of traffic correctly detected by the algorithm to the total number of traffic [5] and indicates how much the output can be trusted. It is calculated by

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{4}$$

4.2.9. Disorder Matrix. It is called a matrix that shows the performance of supervised algorithms. Each column of the matrix shows a sample of the predicted value. This matrix is used to determine the value of evaluation indicators such as precision and accuracy.

5. Creating and Evaluating Machine Learning Models

This paper aims to categorize domain names into risky and legitimate categories based on the created features. It is possible to train with different algorithms and machine learning models to evaluate which algorithm is the most accurate. In this evaluation, logistic regression algorithms, support vector machines, random forest, and decision tree have been studied. Moreover, the technique of analyzing the main components and properties of domain names has been used, which can increase the accuracy of the detection algorithm [1].

First, with random data, the existing dataset is divided into two equal sets for training and testing. The mentioned algorithms were processed and compared for the intended data. In the next step, based on the success rate of the algorithm behavior in terms of correct predictions and positive and negative rates, the results are evaluated which can be read from the deviation matrices. The prediction of classification fields is provided by the Splunk Machine Learning application. These data are processed according to Figures 9–12 with random forest, logistic regression, SVM, and decision tree algorithm. Using the added feature, these algorithms are processed and are in accordance with the SPL commands in Figure 13.

As shown in Figure 14, according to the regression algorithm, the combined error rate of incorrect classification is almost 7%. The values of precision, recall, accuracy, and F1 are 93%. Other algorithms are evaluated according to Figures 15–17, and the performance is evaluated under the same conditions.

As shown in Figure 18 and Table 2, the best performance in terms of the least number of false positives and the best-combined results in terms of the lowest prediction error rate are related to the random forest. However, the decision tree has a similar function. After cross-validation of both algorithms, it could be considered that this algorithm is suitable for this dataset. In this case, the logistic regression algorithm has the lowest accuracy and the highest prediction error rate. Moreover, by adding the PCA feature and comparing Tables 3 and 4, the results show that the prediction of DGA data can be increased by 99.2%.

After the training step of the mentioned models, the properties that have been created in the first part should be calculated. For the data, with the tools in the Splunk base, a random set of sampling domain letters is generated every minute and uses machine learning for the random forest algorithm, decision tree, logistics regression, and SVM in a real search range. The process of correct and incorrect predictions is specified in Figures 19–22 and the results are shown as follows.

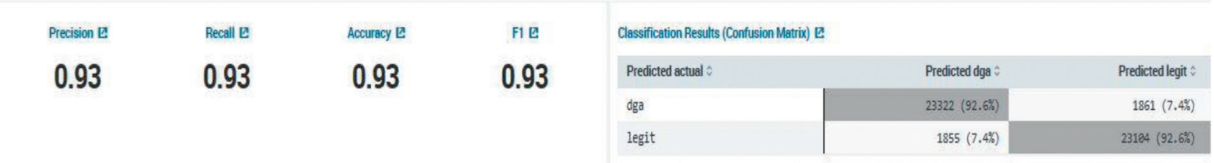
Actual time-based search on random forest, decision tree, logistic regression, and support vector machine is calculated by considering the dataset of the main analysis component and without it. According to Table 5, the analysis time of the decision tree and random forest is lower compared with other algorithms.

The main purpose of applying the dataset used in this research is to reduce the required information for analysis. The list of results shows only the anticipated DGAs and checks whether these DGA algorithms have a security risk or not. Moreover, by editing its code, the graphical interface is customized in which the security analyst is able to provide feedback to the system. If domain name data is reported as legit, the results incorrectly categorized as DGA can be whitelisted. This will cause to reduce the further false positives of the identified DGA domains as shown in Figure 23.

In this mechanism, another level of classification generated can be used in different ways. By manually inserting verified results into the training dataset, the accuracy of the proposed models is improved by continuously training. This helps to keep the models up to date and the training dataset grows larger over time by increasing legitimate domains in whitelists, finally, enlarging the blacklist for accurate matching using the approved categorization. Combining all of these approaches leads to a list of live threats that we can store and modify based on our organizational environment and specifications [18]. The drawn logic is summarized in Figure 24.

6. Evaluation Parameters

The results of the evaluation show that the performance of the algorithm is related to different parameters.

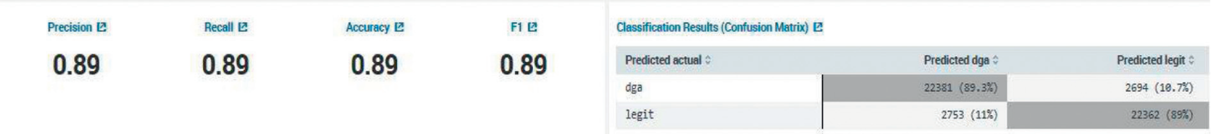


Precision	Recall	Accuracy	F1
0.93	0.93	0.93	0.93

Classification Results (Confusion Matrix)

Predicted actual	Predicted dga	Predicted legit
dga	23322 (92.6%)	1861 (7.4%)
legit	1855 (7.4%)	23104 (92.6%)

Figure 9: Evaluation of a stochastic forest model without using the principal component analysis feature.

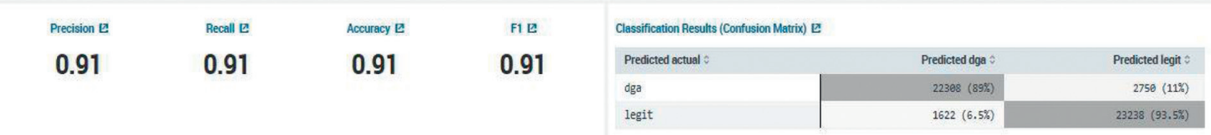


Precision	Recall	Accuracy	F1
0.89	0.89	0.89	0.89

Classification Results (Confusion Matrix)

Predicted actual	Predicted dga	Predicted legit
dga	22381 (89.3%)	2694 (10.7%)
legit	2753 (11%)	22362 (89%)

Figure 10: Results of evaluation based on logistic regression algorithm without using principal component analysis feature.

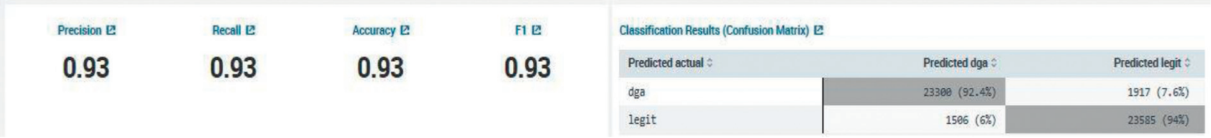


Precision	Recall	Accuracy	F1
0.91	0.91	0.91	0.91

Classification Results (Confusion Matrix)

Predicted actual	Predicted dga	Predicted legit
dga	22308 (89%)	2750 (11%)
legit	1622 (6.5%)	23238 (93.5%)

Figure 11: Value evaluation results based on the SVM algorithm without using the principal component analysis feature.

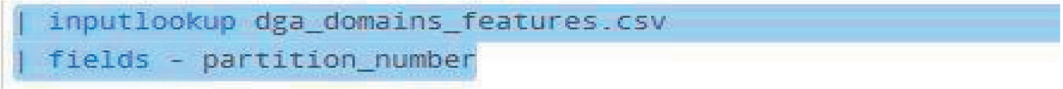


Precision	Recall	Accuracy	F1
0.93	0.93	0.93	0.93

Classification Results (Confusion Matrix)

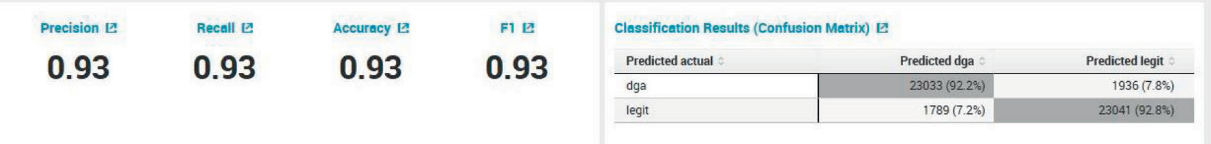
Predicted actual	Predicted dga	Predicted legit
dga	23300 (92.4%)	1917 (7.6%)
legit	1506 (6%)	23585 (94%)

Figure 12: Results of evaluating values based on the decision tree algorithm without using the principal component analysis feature.



```
| inputlookup dga_domains_features.csv
| fields - partition_number
```

Figure 13: SPL commands for evaluating domain names based on specified properties.

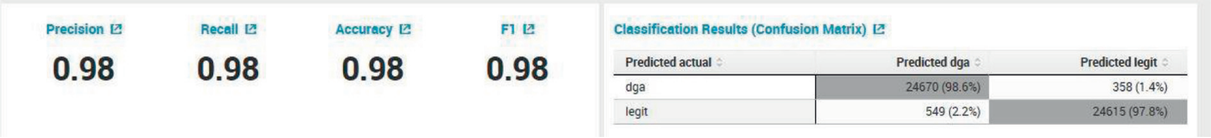


Precision	Recall	Accuracy	F1
0.93	0.93	0.93	0.93

Classification Results (Confusion Matrix)

Predicted actual	Predicted dga	Predicted legit
dga	23033 (92.2%)	1936 (7.8%)
legit	1789 (7.2%)	23041 (92.8%)

Figure 14: Value evaluation results based on logistic regression algorithm using principal component analysis feature.

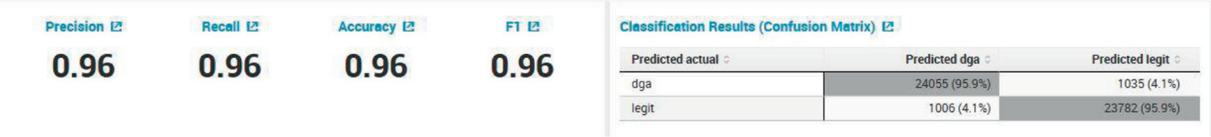


Precision	Recall	Accuracy	F1
0.98	0.98	0.98	0.98

Classification Results (Confusion Matrix)

Predicted actual	Predicted dga	Predicted legit
dga	24670 (98.6%)	358 (1.4%)
legit	549 (2.2%)	24615 (97.8%)

Figure 15: Evaluation of the decision tree model along with the characteristics diagram using the principal component analysis feature.



Precision 0.96

Recall 0.96

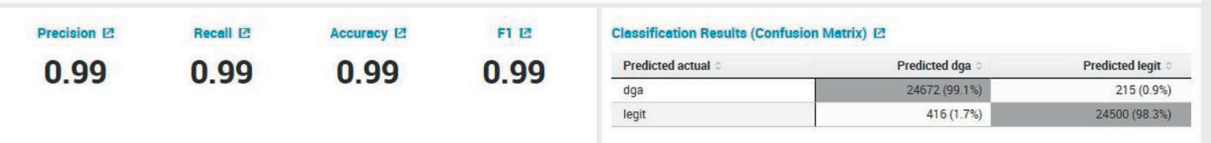
Accuracy 0.96

F1 0.96

Classification Results (Confusion Matrix)

Predicted actual	Predicted dga	Predicted legit
dga	24055 (95.9%)	1035 (4.1%)
legit	1006 (4.1%)	23782 (95.9%)

Figure 16: Evaluation of SVM model with feature diagram using principal component analysis feature.



Precision 0.99

Recall 0.99

Accuracy 0.99

F1 0.99

Classification Results (Confusion Matrix)

Predicted actual	Predicted dga	Predicted legit
dga	24672 (99.1%)	215 (0.9%)
legit	416 (1.7%)	24500 (98.3%)

Figure 17: Evaluation of a random forest model with feature diagram using the principal component analysis feature.

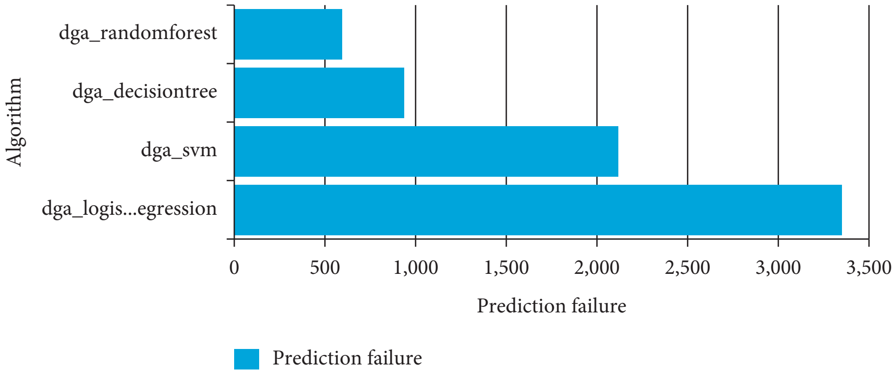


Figure 18: Prediction error rate diagram for DGA botnet detection.

Table 2: Comparison of machine learning models using principal component analysis features.

Machine learning models Criteria	Decision tree	SVM	Random forest	Logistic regression
Accuracy	0.98	0.96	0.99	0.93
Precision	0.98	0.96	0.99	0.93

Table 3: Comparison and evaluation of machine learning models with the main analysis component.

Algorithm name	predicted_dga actual_dga	predicted_dga actual_legit	predicted_legit actual_dga	predicted_legit actual_legit
Decision tree	24702 (99.1%)	376 (0.9%)	217 (1.2%)	24697 (98.3%)
Random forest	24518 (98.6%)	541 (1.4%)	401 (2.2%)	24532 (97.8%)
SVM	23976 (95.9%)	1184 (4.1%)	943 (4.1%)	23889 (95.9)
Logistic regression	23254 (92.2%)	1686 (7.8%)	1665 (7.2%)	23387 (22.6%)

Table 4: Comparison of different algorithms without PCA feature.

Name algorithms	predicted_dga actual_dga	predicted_dga actual_legit	predicted_legit actual_dga	predicted_legit actual_legit
Decision tree	23060 (92.5%)	1879 (7.5%)	1541 (6.2%)	23443 (93.8%)
Random forest	22741 (91.7%)	2066 (8.3%)	1587 (6.4%)	23378 (93.6%)
SVM	22362 (89.7%)	2555 (10.3%)	1885 (7.5%)	23299 (92.5%)
Logistic regression	22381 (89.3%)	2694 (10.7%)	2753 (11%)	22362 (89%)

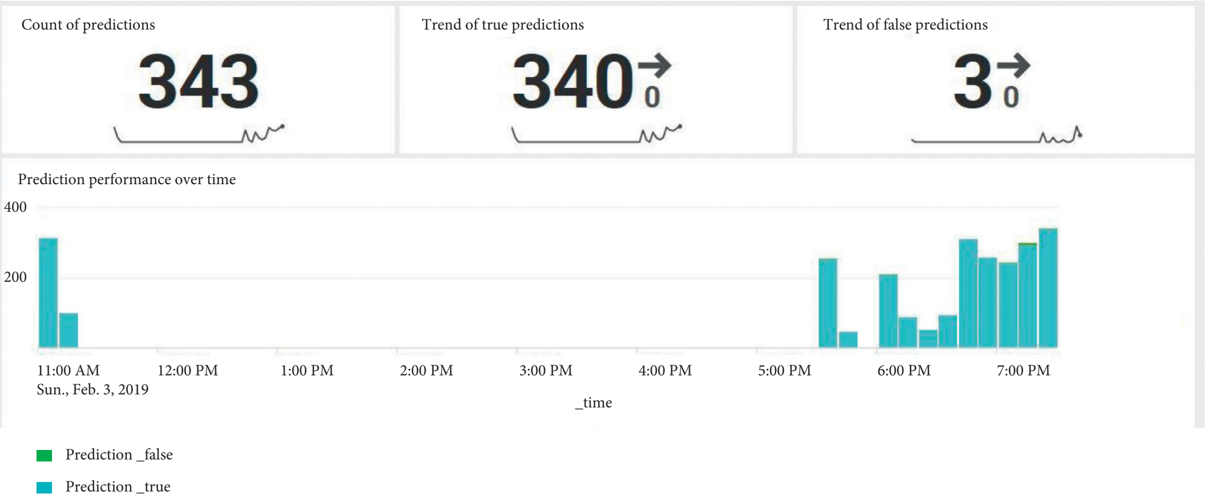


Figure 19: The process of correct and incorrect predictions for the random forest algorithm.

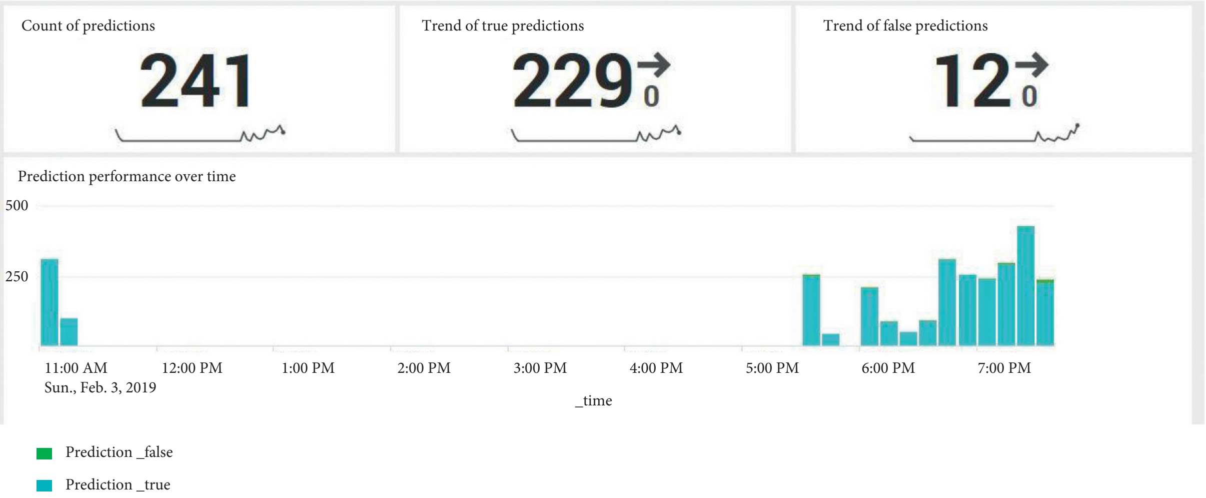


Figure 20: Process of true and false predictions for decision tree algorithm.

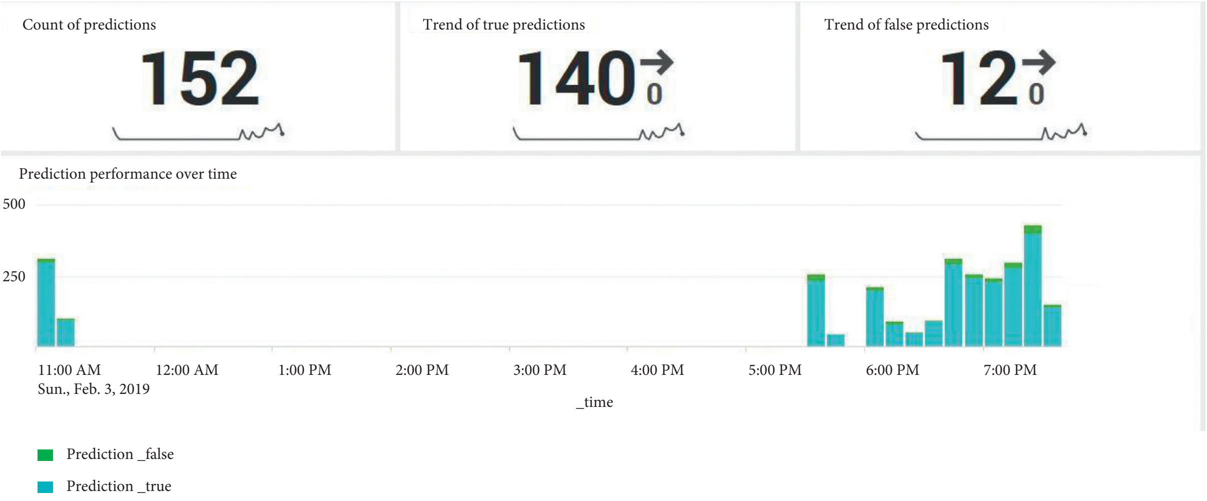


Figure 21: True and false prediction process for logistic regression algorithm.

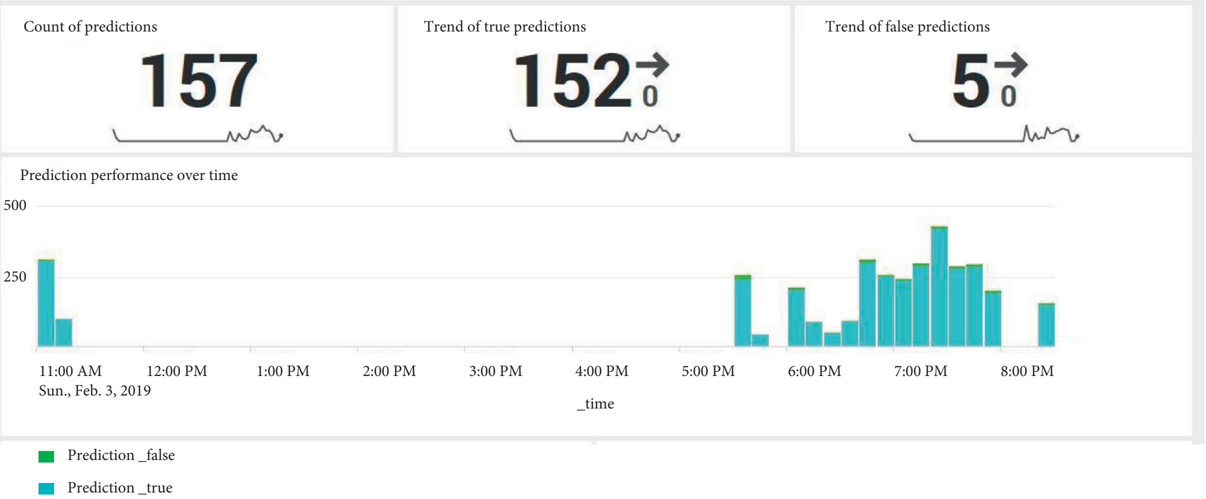
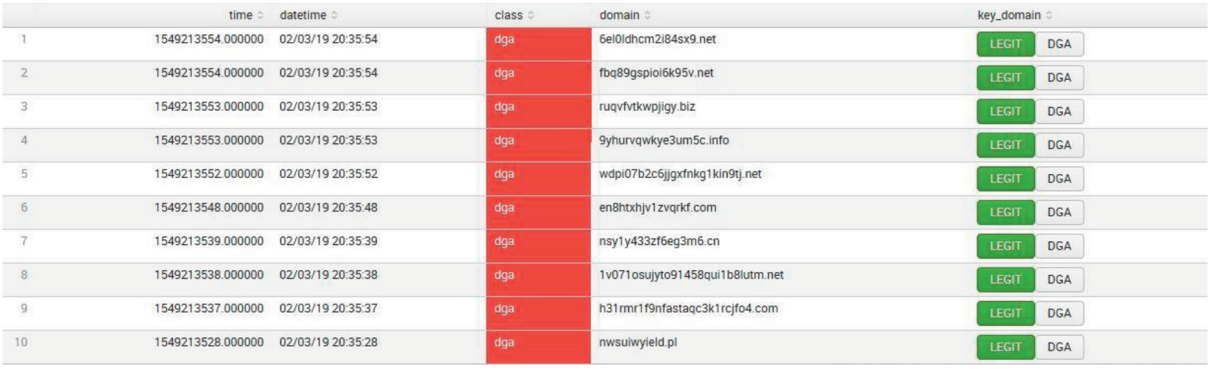


Figure 22: True and false prediction process for SVM algorithm.

Table 5: Comparison of computation time of algorithms with and without the main analysis component dataset.

Name algorithms	Time of algorithms with the main analysis component dataset (sec)	Time of algorithms without the main analysis component dataset (sec)
Decision tree	71.128	75.467
Random forest	69.027	73.158
SVM	81.208	97.621
Logistic regression	77.817	92.237



	time	datetime	class	domain	key_domain
1	1549213554.000000	02/03/19 20:35:54	dga	6el0ldhcm2i84sx9.net	LEGIT DGA
2	1549213554.000000	02/03/19 20:35:54	dga	fbq89gspioi6k95v.net	LEGIT DGA
3	1549213553.000000	02/03/19 20:35:53	dga	ruqvfvtkwpjigy.biz	LEGIT DGA
4	1549213553.000000	02/03/19 20:35:53	dga	9yhurvqwkye3um5c.info	LEGIT DGA
5	1549213552.000000	02/03/19 20:35:52	dga	wdpi07b2c6jjgxfnkg1kin9tj.net	LEGIT DGA
6	1549213548.000000	02/03/19 20:35:48	dga	en8htxhjv1zvqrkf.com	LEGIT DGA
7	1549213539.000000	02/03/19 20:35:39	dga	nsy1y433zf6eg3m6.cn	LEGIT DGA
8	1549213538.000000	02/03/19 20:35:38	dga	1v071osujyto91458qui1b8lutm.net	LEGIT DGA
9	1549213537.000000	02/03/19 20:35:37	dga	h31rmr1f9nfastaqc3k1rcjfo4.com	LEGIT DGA
10	1549213528.000000	02/03/19 20:35:28	dga	nwsuiwyield.pl	LEGIT DGA

Figure 23: DGA detected domains and the editing prediction of domains detected incorrectly.

6.1. Data Simulation and Analysis. To evaluate the performance of botnet domain name classifiers using machine learning algorithms, we use extracted and tagged domain name datasets that contain healthy and infected DTA botnet data.

6.2. Design and Selection of Properties. In the dataset, based on a text-mining approach, we explore domain name strings with n-gram analysis and PCA and identify specific patterns and features that exist in the domain name structure.

6.3. Modeling with Machine Learning Algorithms. We analyze domain letters based on the created features, through machine learning models with logistic regression algorithms, support vector machine, random forest, and decision tree.

6.4. Evaluation and Comparison of Evaluation Criteria. The results of experiments on DGA botnet datasets show that most of the machine learning techniques used to achieve an overall classification accuracy of more than 95%, and

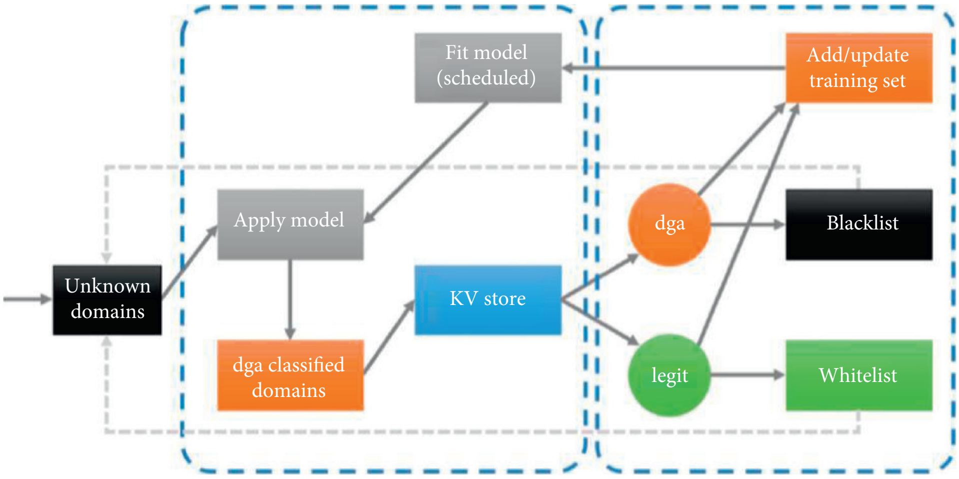


Figure 24: Classification and analysis of DGA domains.

among them, the random forest algorithm, with higher classification accuracy, produces the best results and selecting the dataset feature can have a significant impact on improving the results.

Following achievements are provided to solve the DGA botnet problem in this study.

6.5. Extraction of the Most Optimal Features. Using text mining in identifying control centers and DGA botnet commands is independent of the structure; the extracted features can be identified with high accuracy botnet control centers independent of their structure (centralized, peer-to-peer, and infrastructure).

6.6. Providing Detection Methods. Use machine learning methods and algorithms provided for intelligent identification of botnets. With these algorithms, after a short learning time model, it will be able to automatically and intelligently identify bot-infected domains from normal domains.

6.7. Ability to Be Online. The proposed method is online and can be located next to an online monitoring system to perform DGA domain detection.

7. Conclusion

The new generation of botnets is using the fluxing technique to keep away from the blacklist. Research shows that cybercriminals are increasingly turning on these techniques to evade traditional detection methods. One of the features of DNS traffic is that it can evade random name generation algorithm attacks. This is a new challenge for attackers to improve their DGA bots. In this research, a botnet detection model is presented with a random domain generation algorithm based on machine learning. This model is using domain name query data based on a text-mining model. Therefore, the behavior of botnets that using domain-flux techniques to hide command and control channels was investigated.

The applied machine learning algorithm and text-mining technique to analyze the DNS protocol and identify botnets were also described. For this purpose, extracted and labeled domain name datasets containing clean and infected DGA botnet data were used. In order to explore domain name strings, n-gram analysis and PCA are used to preprocessing data based on a text-mining approach. Some features are existing in the domain name structure. The feature selection method was used to increase the accuracy of machine learning algorithms.

The evaluation results show that the performance of the algorithm varies according to different parameters. Precision indicates the rate of bot traffic compared to the identified bot traffic by the algorithm. The recall parameter also specifies the rate of detecting bot traffic by the algorithm. According to the test results, it can be seen that, among the decision tree, SVM, random forest, and logistic regression algorithms, the logistic regression algorithm has the lowest overall classification accuracy. Furthermore, the random forest algorithm has the highest classification accuracy. However, the differences in the classification accuracy of other algorithms are not large. The decision tree and the SVM algorithm have almost the same overall classification accuracy. The results of the stochastic forest algorithm are significantly better than the decision tree algorithm. However, due to the large number of trees required for training, the training time for the random forest is longer. However, random forest classification training can be done offline. Hence, it does not affect the classification speed during the test period. The random forest machine learning algorithm has the highest overall classification accuracy. It is selected for implementing the proposed botnet detection model.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] X. D. Hoang and Q. C. Nguyen, "Botnet detection based on machine learning techniques using DNS query data," *Future Internet*, vol. 10, no. 5, p. 43, 2018.

[2] K. Alieyan, A. Almomani, A. Manasrah, and M. M. Kadhum, "A survey of botnet detection based on DNS," *Neural Computing and Applications*, vol. 28, no. 7, pp. 1541–1558, 2017.

[3] X. Li, J. Wang, and X. Zhang, "Botnet detection technology based on DNS," *Future Internet*, vol. 9, no. 4, p. 55, 2017.

[4] S. Ryu and B. Yang, "A comparative study of machine learning algorithms and their ensembles for botnet detection," *Journal of Computer and Communications*, vol. 6, no. 5, pp. 119–129, 2018.

[5] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon, "Peer-to-Peer botnets: overview and case study," in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, Cambridge, MA, USA, April 2007.

[6] R. Vinayakumar, K. P. Soman, P. Poornachandran, M. Alazab, and A. Jolfaei, *DBD: Deep Learning DGA-Based Botnet Detection*, pp. 127–149, Macquarie University, Sydney, Australia, 2019, in English.

[7] K. Highnam, D. Puzio, S. Luo, and N. R. Jennings, "Real-time detection of dictionary DGA network traffic using Deep learning," *SN Computer Science*, vol. 2, no. 2, p. 110, 2021.

[8] K. Shinan, K. Alsubhi, A. Alzahrani, and M. U. Ashraf, "Machine learning-based botnet detection in software-defined network: a systematic review," *Symmetry*, vol. 13, no. 5, p. 866, 2021.

[9] A. A. Daya, M. A. Salahuddin, N. Limam, and R. Boutaba, "A graph-based machine learning approach for bot detection," in *Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 144–152, Washington DC, USA, April 2019.

[10] S. Chowdhury, M. Khanzadeh, R. Akula et al., "Botnet detection using graph-based feature clustering," *Journal of Big Data*, vol. 4, pp. 1–23, 2017.

[11] T. D. Tu, C. Guang, and L. Y. Xin, "Detecting bot-infected machines based on analyzing the similar periodic DNS queries," in *Proceedings of the 2015 International Conference on Communications, Management and Telecommunications (ComManTel)*, pp. 35–40, DaNang, Vietnam, December 2015.

[12] P. Drieger, "DGA app for Splunk," 2018, https://splunkbase.splunk.com/app/3559/.

[13] Splunk Inc., "Splunk machine learning toolkit," 2018, https://splunkbase.splunk.com/app/2890/.

[14] Splunk Inc., "Python for scientific computing," 2018, https://splunkbase.splunk.com/app/2881/.

[15] Splunk Inc., "Parallel coordinates—custom visualization," 2018, https://splunkbase.splunk.com/app/3137/.

[16] X. Johnson, "3D scatterplot—custom visualization," 2018, https://splunkbase.splunk.com/app/3138/.

[17] Spamhaus Malware Labs, "Spamhaus botnet threat report 2017," 2018, https://www.spamhaus.org/news/article/772/spamhaus-botnet-threat-report-2017.

[18] Splunk Inc., "Operationalizing-machine-learning-to-detect-malicious-domain," 2018, https://www.splunk.com/en_us/form/operationalizing-machine-learning-to-detect-malicious-domain.html.

[19] D. Acarali, M. Rajarajan, N. Komninos, and I. Herwono, "Survey of approaches and features for the identification of HTTP-based botnet traffic," *Journal of Network and Computer Applications*, vol. 76, pp. 1–15, 2016.