2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress

# Classifying Malicious Domains using DNS Traffic Analysis

Samaneh Mahdavifar, Nasim Maleki, Arash Habibi Lashkari, Matt Broda, Amir H. Razavi

Canadian Institute for Cybersecurity, University of New Brunswick; Bell Canada (BCE), Cyber Threat Intelligence (CTI)
smahdavi@unb.ca, nmaleki@unb.ca, a.habibi.l@unb.ca, matt.broda@bell.ca, amir.razavi@bell.ca

***Abstract*—Malicious domains are one of the major threats that have jeopardized the viability of the Internet over the years. Threat actors usually abuse the Domain Name System (DNS) to lure users to be victims of malicious domains hosting drive-by-download malware, botnets, phishing websites, or spam messages. Each year, many large corporations are impacted by these threats, resulting in huge financial losses in a single attack. Thus, detecting and classifying a malicious domain in a timely manner is essential. Previously, filtering the domains against blacklists was the only way to detect malicious domains, however, this approach was unable to detect newly generated domains. Recently, Machine Learning (ML) techniques have helped to enhance the detection capability of domain vetting systems. A solid feature engineering mechanism plays a pivotal role in boosting the performance of any ML model. Therefore, we have extracted effective and practical features from DNS traffic categorizing them into three groups of lexical-based, DNS statistical-based, and third party-based features. Third party features are biographical information about a specific domain extracted from third party APIs. The benign to malicious domain ratio is also critical to simulate the real-world scheme where approximately 99% of the traffic is devoted to benign. In this paper, we generate and release a large DNS features dataset of 400,000 benign and 13,011 malicious samples processed from a million benign and 51,453 known-malicious domains from publicly available datasets. The malicious samples span between three categories of spam, phishing, and malware. Our dataset, namely CIC-Bell-DNS2021 replicates the real-world scenarios with frequent benign traffic and diverse malicious domain types. We train and validate a classification model that, unlike previous works that focus on binary detection, detects the type of the attack, i.e., spam, phishing, and malware. Classification performance of various ML algorithms on our generated dataset proves the effectiveness of our model, where we achieved the best results for *k*-Nearest Neighbors (*k*-NN) with 94.8% and 99.4% F1-Score for balanced data ratio (60/40%) and imbalanced data ratio (97/3%), respectively. Finally, we have gone through feature evaluation using information gain analysis to get the merits of each feature in each category, proving the third party features as the most influential one among the top 13 features.**

***keywords*— Malicious Domain, DNS, Feature Engineering, Lexical, Statistical, Third Party, Classification**

## I. Introduction

DNS is a hierarchical naming protocol that translates the domain names into Internet Protocol (IP) addresses needed for uniquely identifying computer devices and services over the Internet. This decentralized system has played an essential role in the viability of the Internet for years since it does not require the users to memorize the cumbersome IP addresses of numerous Internet resources. However, DNS communication has been a convenient way for cybercriminals to exploit. The reason is that, unlike other services like HTTP, FTP, or email, controlling DNS traffic is often evaded in enterprises. Due to the significance of DNS service for all applications, most enterprises configure the firewalls to allow all UDP port 53 (used by a DNS server) traffic in. Cybercriminals can easily purchase a domain name for launching various types of malicious activities: to spread malware, such as drive-by download, botnets, and ransomware, to facilitate Command and Control (C&C) communications, to generate phishing websites, and to send spam messages. These cyber-attacks inflict significant financial loss for enterprises. Based on the results of the 2020 Global DNS Threat Report by EfficientIP [1], almost 79% of surveyed organizations have experienced DNS attacks with an average cost of each attack approximated to be nearly $924,000. North America ranks first in terms of attack damages with the highest average cost of DNS attack at $1,073,000.

Many detection systems have been proposed to date to identify malicious domains based on their unique behaviours revealed in the DNS requests/responses [2], [3]. Although analyzing DNS traffic is one of the most promising ways to detect malicious domains, it is not sufficient to identify sophisticated malware. The attackers can utilize modern ways to circumvent detection systems and prevent malicious domains from being easily taken down. For example, Domain Generated Algorithms (DGAs) are programs used by modern botnets to periodically generate a list of domains as rendezvous points with C&C server. Therefore, designing an accurate detection system that relies on not only DNS traffic, but also the structural, linguistic, and biographical properties of a domain is of paramount importance. As explained, a malicious domain could be created for realizing an attacker's intents, such as disseminating malware, sending spam messages, and hosting phishing webpages. Identifying the type of malicious domain can help us to further improve our detection system and devise a proper remediation technique as a preventive measure.

In this paper, we propose to develop and implement a mechanism to detect malicious domains and classify them into one of the categories of malware, spam, phishing, and benign. We develop a feature extractor to generate a comprehensive set of features based on lexical, statistical information of DNS responses, and third party. Our ML-based classification

978-1-6654-2174-4/21/$31.00 ©2021 IEEE
DOI 10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00024

module has been trained and validated using the features extracted from almost 10 GB of DNS traffic. The DNS traffic has been captured while sending HTTP requests to over a million benign and malicious domains. Briefly, our contributions could be specified as follows:

- We provide a methodology for feature engineering of packet captures. We develop 32 clearly defined discriminative features including lexical-based, DNS statistical-based, and third party-based (biographical) features. We then acquire the top 13 features using *infogain* algorithm. Our analysis shows that third party-based features have higher total information gain, thus play a more important role in the classification of the domains.
- We generate and release a large diverse dataset of DNS responses, namely CIC-Bell-DNS2021, which includes 400,000 benign and 13,011 malicious samples processed from benign (over a million) and malicious (51,453) domains and over seven million captured DNS packets. The data ratio of benign/malicious domains is approximately 99/1%, which is to the best of our knowledge, the data ratio in real-world scenarios. Moreover, the malicious domains span between three different categories of malware, phishing, and spam.
- We design a deployment model that identifies malicious domains in real time using a web application and a real-time detection server.
- We achieve state-of-the-art classification results on datasets with imbalanced and balanced data ratio for a range of ML classifiers, namely Support Vector Machine (SVM), *k*-NN, Multi-layer Perceptron (MLP), Gaussian Naive Bayes (GNB), and Logistic Regression (LR) in terms of accuracy, F1-Score, and precision.
- To show the efficacy of our proposed model in real-life on new variants of malicious domains, we conducted the experiments on a validated dataset from a later timeline (December 2020) different than training data (June 2019). We could reproduce the accuracy of the testing phase using SVM, *k*-NN, and GNB classifiers with the highest performance belongs to *k*-NN achieving 98.9% accuracy, 98.9% F1-Score, and 99.0% precision.

The rest of the paper is organized as follows. Related work is discussed in Section II. In Section III, we describe the stages of our proposed model, extracted features, and our deployment model. Section IV discusses the generated dataset and performance analysis. Finally, Section V concludes the paper and outlines some future research directions.

## II. Related Work

In an early study in 2010 [4], a scalable ML classifier was developed to detect phishing websites and maintain Google's phishing blacklist automatically. Their classifier analyzes millions of pages a day, examining the URL and the contents of a page to determine whether or not a page is phishing. In an early study, Choi et al. [5] proposed to use ML for detecting malicious URLs of attack types including spam, phishing, malware, using a variety of features including textual properties, link structures, webpage contents, DNS information, and network traffic. They do experiments on a dataset with 40,000 benign URLs and 32,000 malicious URLs obtained from real-life Internet sources: the accuracy was 98% in detecting malicious URLs and over 93% in identifying attack types. Another approach [6] proposed a method to perform both binary and multi-class URL classification. The class types were benign, malware, phishing, and spam. They developed 42 new features of spam, phishing, and malware URLs. The binary and multi-class dataset was constructed using 49,935 malicious and benign URLs. To evaluate the proposed approach, a set of batch and online ML algorithms were used: One-vs-All SVM, One-vs-One SVM, and multi-class Online Confidence Weighted (CW) Learning. They reported 98.44% average detection accuracy and 1.565% error rate in multi-class setting and 99.86% detection accuracy with an error rate of 0.14% in the binary setting. In [7], Lison et al. proposed a Deep Neural Network (DNN) that can automatically detect whether domain names and IP addresses are benign, malicious, or sinkholes. Evaluation based on a passive DNS dataset demonstrates the effectiveness of the approach as the model can detect 95% of the malicious hosts. The advantage of this method in comparison to traditional reputation lists (white or blacklists) is that it provides predictions in real time and it is less vulnerable to human errors than traditional reputation lists. In another study [8], Jiang et al. applied a Convolutional Neural Network (CNN) model to detect malicious URLs, though, the model is an online method based on character-level DNNs of the URL and DNS strings. Experimental results show that the proposed method outperforms several state-of-the-art baseline methods, in terms of efficiency and scalability. In this paper [9], an Extreme Learning Machine (ELM) method was proposed that was a modern neural network with high accuracy and fast learning speed. They applied ELM to classify malware domain names based on features extracted from multiple resources. This study [10] proposed the automated tracing of malicious websites in a Malware Distribution Network (MDN). It conducts a comprehensive analysis of the total cost involved in visiting websites through automated links and classifies websites as malicious and normal. Palaniappan et al. [11] built a classifier model using the LR classification algorithm and used that classifier to identify benign and malicious domains. In this paper, they not only applied the lexical, web-based, and DNS data for their detection but also applied the IP and domain blacklists to strengthen their detection method. Another study [12] also applied ML algorithms on common features in the detection of malicious domains, such as URL-based, domain-based, and webpage related. They could achieve good accuracy using the Naive Bayes (NB) algorithm. However, the key point in their analysis is that they applied a balanced dataset (half malicious- half benign). Hence, this analysis may not be reliable in the real world. They [13] introduced a lightweight automatic blacklist generator (AutoBLG) framework that can automatically identify new drive-by download URLs. To accel-

erate the process of generating a URL blacklist, they applied ML to reduce the number of URLs to be analyzed after expanding the search space of webpages. AutoBLG consists of three primary components: URL expansion, URL filtering, and URL verification. They illustrated that AutoBLG successfully flags new and previously unknown drive-by-download URLs effectively and efficiently and achieved a higher noise filter rate of 99%. Table I compares the studies based on the method used, type of classification, i.e., multi-class or binary, and the performance result they acquired.

TABLE I
COMPARISON AMONG STUDIES FOCUSING ON MALICIOUS DOMAIN/URL DETECTION (PRC STANDS FOR PRECISION AND ACC STANDS FOR ACCURACY)

| Authors | Method | Result | Class |
|---|---|---|---|
| Whittaker et al. [4] | LR | PRC: 90% | Binary |
| Choi et al. [5] | SVM (Binary) RAkEL & kNN (Multi) | ACC (Binary): 98% ACC(Multi):93% | Multi Binary |
| Patil et al. [6] | One-vs-all SVM One-vs-One SVM Multi Online CW Learning | ACC(Binary): 99.86% ACC(Multi): 98.44% | Multi Binary |
| Lison et al. [7] | DNN | ACC: 95% | Multi |
| Jiang et al. [8] | CNN | 40 mislabeled URLs per 1000 URLS | Binary |
| Shi et al. [9] | ELM | ACC: 96.28% | Binary |
| Choi et al. [10] | AutoLink-Tracer | ACC: 98.08% | Binary |
| Palaniappan et al. [11] | LR | ACC: 60% | Binary |
| Kumar et al. [12] | NB | ACC: 98% | Binary |
| Bo Sun et al. [13] | Bayesian Sets | ACC: 99% | Binary |

## III. PROPOSED METHOD

In this section, we first explain three different categories of discriminative features used in our malicious domain classifier. Then, we describe the stages of our proposed model. Finally, we present a conceptual model of our real-time deployment.

### A. Proposed Features

In this subsection, we explain the extracted features in three categories of lexical, DNS statistical, and third party. First, the captured DNS PCAP file is read and all the domains in the answer section of type A query responses are retained. The field 'rrname' keeps the domain name. Meanwhile, the statistical features are extracted from the structure of the DNS message in a specific packet window. Then, for each captured domain, we extract the lexical and third party features.

*1) DNS Statistical:* DNS statistical features are statistical information computed from the answer section of the DNS responses. The statistical functions are the average of Time-to-Live (TTL), the variance of TTL, distinct number of TTL, domain, IP, Autonomous System Number (ASN), and country in each DNS packet. The features are computed within a sliding window of length $\tau$ with a stride of $s$, as illustrated in Fig 1. As an example, for the TTL feature, the average of the first four TTL values in the sliding window in Fig. 1(a) is calculated as 1625 which is repeated in the first three rows of the table in Fig. 1(b). Then the window is shifted three times down and the next four TTL values are taken for computing the average and so forth.

| Country | ASN | TTL | IP | Domain |
|---|---|---|---|---|
| US | 135 | 1000 | 165.123... | google.com |
| CA | 136 | 2000 | 195.202... | amazon.ca |
| US | 135 | 1500 | 174.204... | appellats.world.com |
| AT | 136 | 2000 | 165.123... | google.com |
| CN | 867 | 1000 | 195.202.... | Xyz-95.cn |
| JP | 923 | 4500 | 205.174... | koitera.com |
| FR | 250 | 3200 | 235.963... | ma-ville.fr |

$s = 3$

$\tau = 4$

(a) Sliding window length= $\tau = 4$, stride= $s = 3$

| No. Unq Country | No. Unq ASN | Avg TTL | Var TTL | No. Unq IP | No. Unq Domain |
|---|---|---|---|---|---|
| 3 | 2 | 1625 | 171875 | 3 | 3 |
| 3 | 2 | 1625 | 171875 | 3 | 3 |
| 3 | 2 | 1625 | 171875 | 3 | 3 |
| 4 | 4 | 2675 | 1716875 | 4 | 4 |
| 4 | 4 | 2675 | 1716875 | 4 | 4 |
| 4 | 4 | 2675 | 1716875 | 4 | 4 |
| 4 | 4 | 2675 | 1716875 | 4 | 4 |

(b) DNS statistical feature values

Fig. 1. Calculating DNS statistical features by sliding window over the CSV file. We set the sliding window size ($\tau$) to 100 and stride ($s$) to 10.

*2) Lexical:* Lexical features help detect malicious domain names since attackers apply different typosquatting and obfuscation methods to mimic the real domain names. In this paper, we have extracted twelve features from each domain elaborated as follows:

**Subdomain:** Most malicious domains have few subdomains to mimic the real domain. For instance, to mimic www.facebook.com, the malicious domain could be in the format of www.facebook.f.com. In this case, the Second-Level Domain (SLD) is f and not facebook. However, people without enough background knowledge may be deceived.

**Top-Level Domain (TLD):** Based on the previous studies [14], most malicious domains are with specific TLDs for a long duration of time, such as .com, .pw, to name a few. The possibility of a URL being malicious is higher when its TLD historically has been used for phishing URLs.

**Second-Level Domain (SLD):** SLD is the most critical part of a domain. Extracting features from SLD can help us get more information regarding the organization that registered the domain name. For example, in www.facebook.com, facebook is the SLD.

**Length:** The length of the domain consists of SLD and subdomains. Because phishing URLs or malicious domains are lengthy based on the previously discovered malicious URLs, considering the length feature is necessary.

**Numeric percentage:** This feature indicates the percentage of numerical characters to the length of the domain.

**Character distribution:** This feature computes the distribution of each letter in the domain.

**Entropy:** This feature is based on the letter distribution and Shannon's entropy formula:

$$p = \frac{letter\_distribution}{domain\_length}, Entropy = \sum_{letter_i} p_i \times \log_2 p_i \quad (1)$$

**N-gram:** This feature extracts the uni-gram, bi-gram, and tri-gram of the domain at the character level.

**Longest meaningful word:** This feature will find the longest word in the domain. To this end, we used a dictionary of words with their frequencies. Then, we apply dynamic programming to infer the location of spaces in the domain.

**Distance from bad words:** There is a blacklist of all suspicious and harmful words. After tokenizing the domain to meaningful words, we get the average of the Levenshtein distance of the domain's words from the blacklist.

**Typosquatting method:** There are generally five Typosquatting types including misspellings, singular versions, plural versions, hyphenations, and common domain extensions. We retrieve the list of 500 top domains from Alexa [15], which are all benign and known domain names. Then, we extract the best matches from the dictionary to our domains, and each match has a score. Based on a threshold on the score, we mark the domain as a typo or not. For example, if our domain is go0gle.com, it will return google.com as the best match with an 85% score; hence, we mark it as a typo.

**Obfuscation method:** To the best of our study, we considered nine different ways by which each domain can be obfuscated. If any of the following methods have been applied by the attacker, we mark the domain name as obfuscated: (1) existing @ in URL, (2) IP obfuscation Decimal 8 bits, (3) IP obfuscation Decimal 32 bits, (4) IP obfuscation Octal 8 bits, (5) IP obfuscation Octal 32 bits, (6) IP obfuscation Hexadecimal 8 bits, (7) IP obfuscation Hexadecimal 32 bits, (8) detecting IDN: suspicious domain name may be encoded with Unicode or international domain names encoded with Punycode, and (9) shortened URLs: to confirm if a URL has been shortened or not, we send a request to see if it gets redirected to the real URL or not.

*3) Third Party:* The third party features are extracted from two third party sources, i.e., Whois and Alexa rank and they contain the biographical properties of a domain. Table II depicts the final 32 DNS features.

TABLE II
LIST OF DNS-BASED FEATURES

| Feature | Feature Name | Description |
|---|---|---|
| | | **Lexical** |
| F1 | Subdomain | Has sub-domain or not |
| F2 | TLD | Top-level domain |
| F3 | SLD | Second-level domain |
| F4 | Len | Length of the domain and subdomain |
| F5 | Numeric percentage | Counts the number of digits in the domain and subdomain |
| F6 | Character distribution | Counts the number of each letter in the domain |
| F7 | Entropy | Entropy of letter distribution |
| F8 | 1-gram | 1-gram of the domain in letter level |
| F9 | 2-gram | 2-gram of the domain in letter level |
| F10 | 3-gram | 3-gram of the domain in letter level |
| F11 | Longest word | Longest meaningful word in SLD |
| F12 | Distance from bad words | Computes average distance from bad words |
| F13 | Typos | Typosqautting |
| F14 | Obfuscation | Max value for URL obfuscation |
| | | **DNS Statistical** |
| F15 | Unique country | The number of distinct country names in window $\tau$ |
| F16 | Unique ASN | The number of distinct ASN values in window $\tau$ |
| F17 | Unique TTL | The number of distinct TTL values in window $\tau$ |
| F18 | Unique IP | The number of distinct IP values in window $\tau$ |
| F19 | Unique domain | The number of distinct domain values in window $\tau$ |
| F20 | TTL mean | The average of TTL in window $\tau$ |
| F21 | TTL variance | The variance of TTL in window $\tau$ |
| | | **Third Party** |
| F22 | Domain name | Name of the domain |
| F23 | Registrar | Registrar of the domain |
| F24 | Registrant name | The name the domain has been registered |
| F25 | Creation date time | The date and time the domain created |
| F26 | Emails | The emails associated to a domain |
| F27 | Domain age | The age of a domain |
| F28 | Organization | What organization it is linked to |
| F29 | State | The state the main branch is |
| F30 | Country | The country the main branch is |
| F31 | Name server count | The total number of name servers linked to the domain |
| F32 | Alexa rank | The rank of the domain by Alexa |

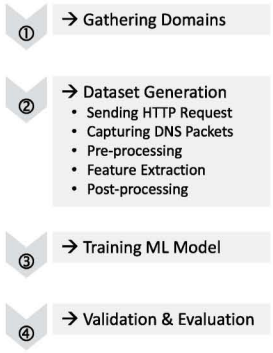


Fig. 2. Stages of the proposed model

### *B. Proposed Model*

Fig. 2 shows four main stages of our proposed model we followed to detect malicious domains and classify them into one of the categories of malware, spam, phishing, and benign. The first stage, which is gathering domains, focuses on collecting a large corpus of benign and malicious domains from Majestic Million [16], OpenPhish [17], PhishTank [18], DNS-BH [19], malwaredomainlist [20], and jwspamspy [21] spanning between four different categories of benign, phishing, malware, and spam.

In the second stage, dataset generation, we send HTTP requests using a Python script to the collected domains and the related network packets are captured using Wireshark. The detailed steps are illustrated in Fig. 3. First, all the domains are retrieved from the client's local database (steps 1 and 2). Then, an HTTP request is sent to each domain's web server through the Internet and the OK (200) response is received back in steps 3 and 4. Meanwhile, the DNS packets are dumped and the domains with a timed-out request are just ignored. To

avoid long waiting times, we set the timeout parameter to two seconds.

After capturing the DNS packets, we pre-process the captured packets by keeping only type A query responses. We discard the queries and responses of type AAAA, CNAME, and SOA. We then apply our DNS feature extractor to extract 32 final DNS features from the captured packets. After that, all the features are post-processed. The categorical features namely, *sld, emails, domain name, country, registrar, state, registrant name, longest word, organization, tld* are transformed and represented as continuous values. Average of *n-gram* frequencies, *char distribution, typos, distance from bad words* were computed. Also, *creation date time* and *domain age* are converted to seconds and years, respectively. The rest of the numerical features remain unchanged. Besides, we calculate the maximum value of the nine obfuscation methods and merge them all into one feature of obfuscation. Finally, we create a CSV file of all the post-processed features values.

The third stage trains the ML model, and the fourth stage validates and evaluates the model using fold cross-validation.

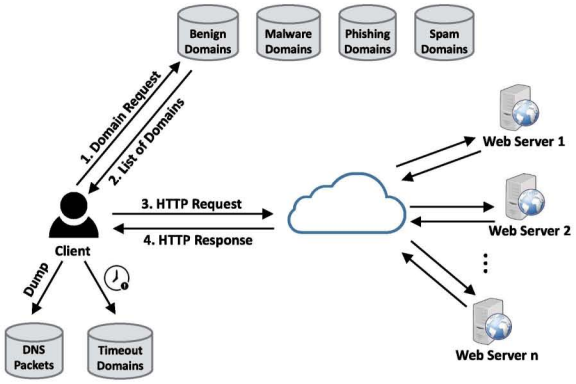


Fig. 3. DNS packet capture

### C. Real-time Deployment

Designing an efficient and applicable vetting system that identifies malicious domains in real time is of paramount importance. To this end, we provide a scheme of our deployment model that predicts the label of a domain in real time through a web application as the front end and a real-time detection server as the back end (Fig. 4). First, the user submits a domain to the web client. The request is forwarded to the real-time server where an HTTP request is sent to the domain, the associated DNS packets are captured, and 32 features are derived from the packets. Then, the trained model predicts the category of the domain and returns the prediction. In the meantime, the new arrival data samples are reported to the offline server which holds a copy of the ML model. The offline server needs to work with data streams and thus it applies incremental learning and hyperparameter optimization. In our scheme, incremental learning realizes by re-training the same model and tunes it according to new arrival data samples. Since incremental learning learns each new data sample individually, the whole process requires low computational cost. In the last step, the classifier and dataset residing on a real-time detection server are updated accordingly to serve coming data streams in the future.

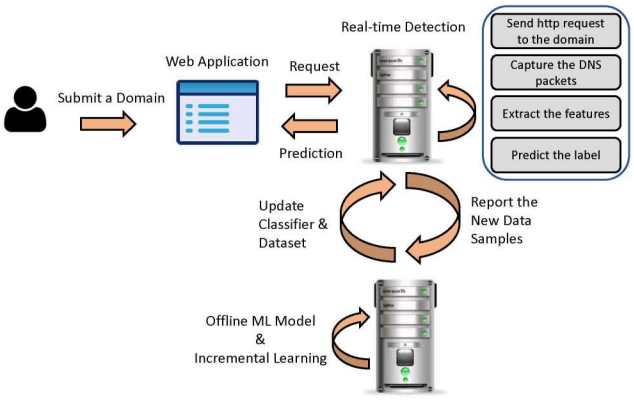


Fig. 4. Real-time prediction

## IV. Experiments and Analysis

In this section, we evaluate the effectiveness of our proposed traffic-based DNS features by leveraging common ML algorithms.

### A. Dataset

One of the contributions of this paper is to generate a large DNS features dataset based on the malicious and benign domains. As explained in Section III, we sent HTTP requests to the gathered domains and the associated packets were captured. Then the captured packets were pre-processed, and the proposed features were extracted from the packets using our developed DNS-based feature extractor. Finally, the features were post-processed to create the final dataset. We make our dataset (CIC-Bell-DNS2021) publicly available for further research improvements in this area[1]. We managed to collect more than one million domains from various sources falling under four different categories of malware, spam, phishing, and benign domains. All the domains have been collected between May 2019 to June 2019 and later updated with domains from December 2020 for further validation. Each domain category is briefly explained as follows:

*1) Malware Domains:* Malware category refers to the domains that have been previously identified to generate any general type of malware including drive-by download, DGA-based botnets, Distributed Denial of Service (DDoS) attacks, and spyware. The malware domains were collected from DNS-BH [19] and malwaredomainlist [20].

[1]https://www.unb.ca/cic/datasets/dns-2021.html

*2) Spam Domains:* Spammers employ different ways to find valid email addresses for sending bulk emails. Dictionary harvest attack is one of the common ways to seek a valid email address by randomly sending mail to widely used mailbox names for a domain, such as info@example.com, admin@example.com, and support@example.com. We have obtained Spam domains from jwspamspy [21] which works as an e-mail spam filter for Microsoft Windows.

*3) Phishing Domains:* Phishing domains imitate the looks of legitimate websites and leverage social engineering techniques to trick users into clicking the malicious link. Upon clicking the fake link through email or SMS, the user is directed to an imposter website asking for the user's login credentials and private information. In our dataset, the phishing domains were collected from OpenPhish [17] and PhishTank [18] which are collaborative phishing verification websites where the users submit the phishing data, and the community users vote for it.

*4) Benign Domains:* All the domains that are not in the above categories are deemed to be benign. We gathered all benign domains from Majestic Million [16].

Table III shows the breakdown of the dataset in terms of the number of collected domains (#original domain) in each category of malware, spam, phishing, and benign. After sending HTTP requests to each domain, some of the domains do not respond, e.g., C&C servers that are not alive anymore. The remaining domains that respond OK (200) are logged, and associated DNS packets are dumped. In Table III, the domains and packets that have been successfully processed are identified with columns '#domains processed' and '#packets processed', respectively.

TABLE III
STATISTICS OF THE DOMAINS DATASET

| Category | #Original Domains | #Domains Processed | #Packets Processed |
|---|---|---|---|
| **Malware** | 26,895 | 9,432 | 182,266 |
| **Spam** | 8,254 | 1,976 | 61,046 |
| **Phishing** | 16,307 | 12,586 | 95,492 |
| **Benign** | 988,667 | 500,000 | 6,907,719 |

Table IV provides a comparison among our dataset with other datasets employed in Section II based on general specifications of the dataset and features aspects. In the table, lexical features are used for determining any distinguishable patterns in a URL or domain which might be a clue for any suspicious source. The second category of features is either extracted from a third party like Alexa rank or are referred to any biographical information of a domain, such as the country, state of the domain, or the date the domain has been created. Domain features are mostly DNS information required for detecting malicious websites, i.e., ASN, name server count, and resolved IP count. The last category of features is based on client-side code in the webpage that can be used to detect the injected malicious code into the webpage content.

Our dataset is the most recent with various categories of malicious domains namely malware, phishing, and spam. It is the only public dataset with the approximate benign/malicious data ratio of 99/1%, closest to the real-world data ratio. Furthermore, it comprises 32 discriminative features classified into three groups: lexical, third party, DNS statistical that encompass all the diversified properties of a malicious domain.

### B. Experimental Results

*1) Setup:* We implemented our proposed malicious domain detection system using a server with 8 core processor Intel Xeon E5-2609 v4 @ 1.70GHz and 64 GB RAM. We leveraged Python for feature extraction and Sklearn library [22] for implementing ML algorithms. We set sliding window size ($\tau$) to 100 and stride ($s$) to 10. Moreover, we replace nan (missing) values with zero in the input data.

*2) Analysis:* In the analysis step, we applied a set of ML algorithms on two subdivisions of our dataset, namely balanced and imbalanced. First, we balanced the number of spam, malware, and phishing datasets each having 4,337 samples to generate a total of 13,011 malicious samples. For the benign samples, we created two categories of 20,000 and 400,000 samples for the balanced and imbalanced subdivisions of our dataset, respectively. Thus, we end up with a data ratio of $60/40\%$ (benign/malicious) for a balanced dataset and $97/3\%$ (benign/malicious) for an imbalanced dataset.

We compare the classification results of multiple ML algorithms in Table V. We used stratified 5-fold cross-validation from the Sklearn library which returns stratified folds for training and testing data by preserving the percentage of samples for each class. We shuffled each class's samples before splitting them into train and test bins. As illustrated in Table V, for both $60/40\%$ and $97/3\%$ data ratio, $k$-NN significantly outperforms SVM, $k$-NN, MLP, GNB, and LR in terms of accuracy, F1-Score, and precision. It achieves $94.8\%$ and $99.4\%$ F1-Score for balanced and imbalanced datasets, respectively. For $97/3\%$ data ratio, the gaps between classification results are trivial where GNB ranks second among other classifiers with a negligible difference (almost $0.8\%$ lower for accuracy) from $k$-NN. Overall, the results on the imbalanced dataset are superior to ones on the balanced dataset due to a higher ratio of benign samples that might influence the average classification measures. However, since we have used stratified sampling, we can make sure we have provided more uniformity during training by removing the variance of the proportions inside bins. The configurations of all the classifiers are shown in Table VI.

We applied the feature selection algorithm in Weka, i.e., *InfoGainAttributeEval*, to obtain the top 13 features, out of the underlying dataset. We chose the top 13 since there was a large gap of +0.625 between features ranks 13 and 14. The search method was set to ranker and the evaluation model was customized to be 10-fold cross-validation. Table VII depicts the average merit for each feature ordered by their ranks.

TABLE IV
DATASET COMPARISONS

| General Specifications of Datasets | | | | | | | Features | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Authors | Collection Year | Publish Year | Variety | Public Dataset | Real Data Ratio | Size | Lexical | Third Party | Domain | Web Page |
| Whittaker et al. [4] | 2009 | 2010 | Benign, phishing | | ✓ | 10,783,820 benign, 120,651 phishing | – | 2 | 5 | 3 |
| Choi et al. [5] | 2011 | 2011 | Benign, malware, phishing, spam | | | 40,000 benign, 32,000 malicious | 10 | 15 | 15 | 13 |
| Patil et al. [6] | 2017 | 2018 | Benign, malware, phishing, Spam | | | 26,041 benign, 23,894 malicious | 65 | – | 18 | 34 |
| Lison et al. [7] | 2010 | 2011 | Benign, malicious, sinkhole | | | 38,811,436 benign, 2,903,996 malicious, 14,858 sinkhole | – | 2 | 27 | – |
| Jiang et al. [8] | NA | 2018 | Benign, malicious | | ✓ | 6,000,000 benign, 1,000,000 malicious | 256 | – | – | – |
| Shi et al. [9] | NA | 2018 | Benign, malicious | | | 12,096 benign, 38,915 malicious | 3 | 2 | 4 | – |
| Choi et al. [10] | NA | 2019 | Benign, malicious | | | 10,000 benign, 10,000 malicious | – | 1 | 4 | – |
| Palaniappan et al. [11] | NA | 2020 | Benign, malicious | | | 20,000 | 4 | 4 | 4 | 5 |
| Sun et al. [13] | NA | 2016 | Benign, malicious | | | 10,000 benign, 6 million malicious | – | – | – | 19 |
| Kumar et al. [12] | NA | 2020 | Benign, phishing | | | 57,000 benign, 57,000 malicious | 22 | 4 | – | – |
| CIC-Bell-DNS2021 | 2020 | 2021 | Benign, malware, phishing, spam | ✓ | ✓ | 400,000 benign, 13,011 malicious | 14 | 11 | 7 | – |

TABLE V
COMPARING CLASSIFICATION METRICS % OF A SET OF ML ALGORITHMS

| | Accuracy | | F1-Score | | Precision | |
|---|---|---|---|---|---|---|
| **Algorithm** | **60/40** | **97/3** | **60/40** | **97/3** | **60/40** | **97/3** |
| **SVM** | 61.2 | 97.0 | 47.1 | 95.4 | 50.1 | 95.0 |
| **$k$-NN** | **94.8** | **99.4** | **94.8** | **99.4** | **94.9** | **99.4** |
| **MLP** | 72.2 | 96.8 | 64.6 | 95.3 | 67.8 | 93.8 |
| **GNB** | 78.2 | 98.6 | 75.0 | 98.4 | 85.0 | 99.0 |
| **LR** | 76.9 | 97.8 | 75.4 | 97.6 | 80.7 | 97.8 |

TABLE VI
ML CLASSIFIERS USED IN THE COMPARISON

| Classifier | Configuration |
|---|---|
| SVM | decision_function_type=ovo, kernel=rbf |
| $k$-NN | n_neighbors=3 |
| MLP | hidden_layer_size=(30,20,10), activation=relu, solver=adam, learning_rate_init=0.001 |
| GNB | var_smoothing=$1e^{-9}$ |
| LR | solver=lbfgs, penalty=$l_2$ |

TABLE VII
FEATURE AVERAGE MERIT

| Rank | Average Merit | Feature |
|---|---|---|
| 1 | 0.227 +- 0 | Registrant name |
| 2 | 0.216 +- 0 | Organization |
| 3 | 0.213 +- 0 | Country |
| 4 | 0.212 +- 0 | Emails |
| 5 | 0.211 +- 0 | State |
| 6 | 0.205 +- 0 | TLD |
| 7 | 0.196 +- 0 | Registrar |
| 8 | 0.194 +- 0 | Domain name |
| 9 | 0.192 +- 0.005 | TTL variance |
| 10 | 0.182 +- 0.004 | TTL mean |
| 11 | 0.18 +- 0 | SLD |
| 12 | 0.161 +- 0 | Unique ASN |
| 13 | 0.155 +- 0.001 | Longest word |

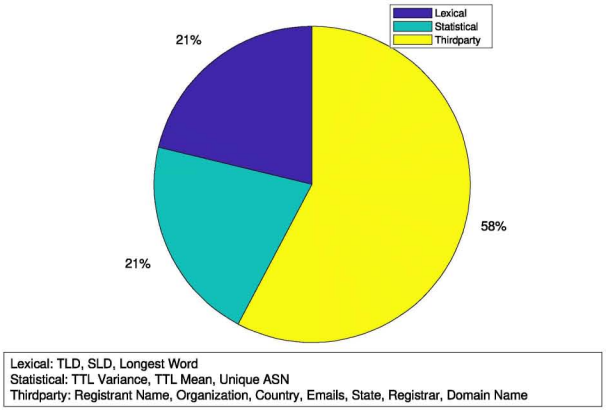


Fig. 5. Information gain of each feature category

Third party features had higher total information gain, 58%, while lexical and statistical acquired the same share of information gain each 21% average merit, as shown in Fig. 5. Hence, it is concluded that third party features are the most important feature category in classifying malicious domains. They indicate the biographical information of a domain mostly obtained from Whois, Alexa rank, or Google rank. Third party information could be a reliable indication of the legitimacy of a domain since they contain up-to-date global registered domain information and rankings of millions of websites.

*3) Validation:* Although our model demonstrates promising classification results, there might be still uncertainties whether it can detect new malicious domains different from training data or from a later timeline. If we only rely on testing on split data set from the same timeline, we are likely to get domains associated with the same malware types and the results will probably be far from how the model will perform on future data. To prove so, we selected 50 malicious domains from PhishTank that have been validated as phishing domains in late 2020. For the benign domains, we picked the 50 top domains from Alexa. For all the new malicious and benign domains, we followed the same procedure in Section III. Then, we applied the ML classifiers likewise the training and testing

stages and produced the results. As shown in Fig. 6, three classifiers SVM, *k*-NN, and GNB demonstrate high detection performance in terms of accuracy, F1-Score, and precision with the highest is *k*-NN achieving 98.9% accuracy, 98.9% F1-Score, and 99.0% precision. Since we do not have a large number of samples in the validation step, MLP apparently does not show acceptable results.

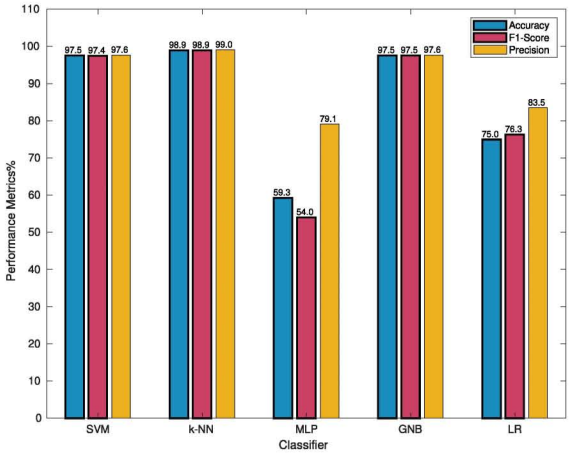


Fig. 6. Validation of model using new malicious domains

## V. Conclusion

In this paper, we proposed an effective and applicable malicious domain classification system based on three categories of features namely, lexical, DNS statistical, and third party acquired by deep inspection of DNS traffic. The proposed detection system achieves state-of-the-art results on several ML techniques. We have shown that third party features are the most important category of features in the classification process with a total of 58% information gain among the top 13 features. We have also released a large dataset of domains (CIC-Bell-DNS2021) containing a million benign and 51,453 malicious domains (malware, phishing, spam) that highly resembles the real-world setting of malicious to benign traffic ratio (1/99)%. We have extracted 32 features from 400,000 benign and 13,011 malicious DNS responses captured from over seven million DNS packets. In the future, we are planning to enrich our feature set by adding more stateful/stateless features.

## References


[1] "Dns attacks are widespread, damaging, and increasingly hitting cloud: 2020 global dns threat report," https://continuitycentral.com/index.php/news/technology/, Accessed August 2020.

[2] S. Torabi, A. Boukhtouta, C. Assi, and M. Debbabi, "Detecting internet abuse by analyzing passive dns traffic: A survey of implemented systems," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 4, pp. 3389–3415, 2018.

[3] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through dns data analysis," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–36, 2018.

[4] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," 2010.

[5] H. Choi, B. B. Zhu, and H. Lee, "Detecting malicious web links and identifying their attack types." *WebApps*, vol. 11, no. 11, p. 218, 2011.

[6] D. R. Patil and J. B. Patil, "Feature-based malicious url and attack type detection using multi-class classification." *ISeCure*, vol. 10, no. 2, 2018.

[7] P. Lison and V. Mavroeidis, "Neural reputation models learned from passive dns data," in *IEEE Int. Conference on Big Data*. IEEE, 2017, pp. 3662–3671.

[8] J. Jiang, J. Chen, K.-K. R. Choo, C. Liu, K. Liu, M. Yu, and Y. Wang, "A deep learning based online malicious url and dns detection scheme," in *Int. Conference on Security and Privacy in Communication Systems*. Springer, 2017, pp. 438–448.

[9] Y. Shi, G. Chen, and J. Li, "Malicious domain name detection based on extreme machine learning," *Neural Process. Lett.*, vol. 48, no. 3, pp. 1347–1357, 2018.

[10] S.-Y. Choi, C. G. Lim, and Y.-M. Kim, "Automated link tracing for classification of malicious websites in malware distribution networks." *J. Inf. Process. Syst.*, vol. 15, no. 1, 2019.

[11] G. Palaniappan, S. Sangeetha, B. Rajendran, S. Goyal, B. Bindhumadhava *et al.*, "Malicious domain detection using machine learning on domain name features, host-based features and web-based features," *Procedia Computer Science*, vol. 171, pp. 654–661, 2020.

[12] J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran, and B. Bindhumadhava, "Phishing website classification and detection using machine learning," in *Int. Conference on Computer Communication and Informatics*. IEEE, 2020, pp. 1–6.

[13] B. Sun, T. Takahashi, L. Zhu, and T. Mori, "Discovering malicious urls using machine learning techniques," in *Data Science in Cybersecurity and Cyberthreat Intelligence*. Springer, 2020, pp. 33–60.

[14] "Phishing activity trends report," https://docs.apwg.org//reports/apwg_trends_report_q1_2019.pdf, Accessed May 2021.

[15] "Alexa," https://www.alexa.com/, Accessed Dec. 2020.

[16] "The majestic million," https://majestic.com/reports/majestic-million, Accessed June 2019.

[17] "Openphish," https://openphish.com/feed.txt, Accessed June 2019.

[18] "Phishtank," https://www.phishtank.com/developer_info.php, Accessed June 2019.

[19] "Dns-bh–malware domain blocklist by riskanalytics," (https://www.malwaredomains.com/?page_id=66), Accessed June 2019.

[20] "Malware domain list," http://www.malwaredomainlist.com/forums/index.php?topic=3270.0, Accessed June, 2019.

[21] "Spam domain list," https://joewein.de/sw/bl-text.htm, Accessed June 2019.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.