Annals of Telecommunications (2024) 79:457–473
https://doi.org/10.1007/s12243-024-01043-3

A review on lexical based malicious domain name detection methods

Cherifa Hamroun[1,2] · Ahmed Amamou[3] · Kamel Haddadou[3] · Hayat Haroun[2] · Guy Pujolle[1]

Received: 27 January 2023 / Accepted: 21 May 2024 / Published online: 13 June 2024

Abstract
Nowadays, domain names are becoming crucial digital assets for any business. However, the media never stopped reporting phishing and identity theft attacks held by third-party entities that rely on domain names to mislead Internet users. Thus, Palo Alto Networks revealed in their studies 20 largely cyber-squatted domain names targeting popular brands. Based on their behavior, domain names appear in public lists that objectively evaluate their reputation. Blacklists contain domain names that have previously committed suspicious acts, whereas whitelists include the most popular and trustworthy domain names. For a long time, this listing technique has been used as a reactive approach to counter domain name-based attacks. However, it suffers from the limitation of responding late to attacks. Nowadays, techniques tend to be much more proactive. They operate before any attack occurs. As part of the CSNET conference, we published a short paper that describes a plethora of domain name attacks and their associated detection techniques using their lexical features (Hamroun et al. 2022). In this paper, we present an extended version of the original one which discusses the previously mentioned points in more detail and adds some elements of understanding when it comes to malicious domain name detection. Hence, we provide a literature review of malicious domain name detection techniques that use only the lexical features of domain names. These features are available, privacy-preserving, and highly improve detection results. The review covers recent works that report relevant performance categorized according to a new taxonomy. Moreover, we introduce a new criterion for comparing all the existing works based on targeted maliciousness type before discussing the limitations and the newly emerging research directions in this field.

Keywords Malicious · Domain names · Lexical analysis · Cybersquatting · Predictive methods

1 Introduction

A domain name is a unique representative name that associates an IP (Internet Protocol) address with an intelligible and easily memorable word sequence. According to the domain name system (DNS), domain name character sequences are structured under a tree structure of suffixes called the domain name space. The root of this tree is the domain represented by a zero-length label. The point character "." is used in domain names to divide hierarchical levels. The parts located between the points are called labels. The right-most label is the Top-Level Domain (TLD), for example, the ".com" TLD. The domain directly to the left of a TLD is called the Second-Level Domain (2LD or SLD), for example: "google.com." The Fully Qualified Domain Name (FQDN) identifies a unique resource record in the domain name space, for example, the website "www.google.com" [2]. With the evolution of the Internet and its protocols, TLDs provide generic information purely indicative of the service associated with the domain name. Hence, they are divided into two types of services: extensions from a given country called ccTLDs (like .dz for Algeria or .fr for France) or generic extensions, gTLDs (like .com or .net). However, recent years have seen the emergence of other, more specific types of extension, such as legacy gTLDs, nTLDs, and penny TLDs. Nowadays, domain names are crucial digital assets for any organization that wants to build a powerful digital image

✉ Cherifa Hamroun
hc_hamroun@esi.dz

Ahmed Amamou
ahmed@gandi.net

Kamel Haddadou
kamel@gandi.net

Hayat Haroun
bh_haroun@esi.dz

Guy Pujolle
Guy.Pujolle@lip6.fr

1 Laboratoire d'Informatique de Paris 6, Paris, France

2 École Nationale Supérieure d'Informatique, Algiers, Algeria

3 GANDI, Paris, France

on a global scale. Therefore, attackers undertake illicit acts that target organizations and Internet users via their domain names. As a result, the threat is omnipresent on the Internet. Phishing and identity theft attacks are increasing and indiscriminately targeting the victims. Since the cybersecurity world is facing an emerging challenge, efforts are multiplying to address it. However, researchers differ in what they consider a "malicious" domain name. In this work, we define a malicious domain name as any domain name involved in any attack that does not directly target the domain name system (DNS). In other words, we consider only attacks that target Internet users using social engineering and cybersquatting techniques. The limitation of the targeted attacks enables us to present a more concise state-of-the-art view of domain name detection methods based only on lexical features since attackers take advantage of the structural composition of the domain name to alter its behavior.

This paper is structured as follows:

- First, we examine all the research axes of the malicious domain name detection field to provide a global vision of the context and motivations that led us to consider this area of research rather than other promising ones.
- Second, we describe in detail domain name hijacking means that can be considered while designing a detection technique that efficiently encounters: phishing, spam, or commandment and control (C&C) attack that targets a specific domain name.
- Third, we synthesize all state-of-the-art detection methods and existing works related to them. We also briefly explain the functioning of every detection system and report its performance.
- Finally, we compare their performance before discussing their limitations and the promising research directions to be further explored in the future.

As mentioned earlier in the abstract, this work is an extended version of a previous paper published during the CSNet 2022 conference [1].

2 Context and motivations

The emergence of domain names popularized the use of the Internet. As domain names contributed to the Internet revolution, they quickly became an attack infrastructure that targets various institutions. The recently undertaken attacks by attackers aim to alter domain names' behavior or ruin the reputation of their holders. Cyber analysts use remaining attackers' traces for tracking and forecasting these attacks. However, a domain name has several characteristics that define its behavior. We introduce in Fig. 1 a taxonomy of malicious domain names detection methods based on these characteristics. We classify these methods under three general categories:

- **Context-Aware methods:** that predict the nature of a domain name using its network traffic characteristics.
- **Context-Free methods:** that predict the nature of a domain name using its lexical[1] composition characteristics.
- **Hybrid methods:** that combine network-based and lexical characteristics of the domain name to predict its behavior in the short, medium, or long term.

Context-aware characteristics are network-based. Thus, they include attributes such as DNS resource records, WHOIS objects, and SSL certificates of the domain name [3]. These characteristics are hard to acquire because packet inspection and network analysis of domain names compromise user privacy. However, private data is seen as the new gold in this rapidly growing technological era. Therefore the tightening of privacy and confidentiality policies is a piece of evidence. Accordingly, malicious domain names are often recognized only by context-free characteristics. These characteristics exploit the character correspondence and the content of domain names' character sequences [3]. Domain names are supposed to be easily memorizable or mimic an easily recognizable sequence of characters since the main objective of any DNS service is to provide a Domain-IP association. Therefore, researchers suggest replacing privacy-affecting solutions with more privacy-oriented ones. Since linguistic property-based methods are anonymous and do not require any contextual information about users, they seem to be an attractive alternative [4]. These techniques achieve relevant (and often better) detection results at a lower cost. Consequently, we devote this work to presenting a literature review of context-free detection methods.

3 Research methodology

As stated earlier in this paper, we focus on studies that propose new context-free feature-based detection methods. Hence, we followed a meticulous research strategy to find a considerable amount of papers that report reproducible results. The following points detail the stated methodology:

1. We first managed to gather all the surveys and reviews that dealt with domain name detection methods indiscriminately using a set of keywords: detection, malicious, domain names.

[1] Also known as context-free, textual, semantic, statistical, or linguistic features.

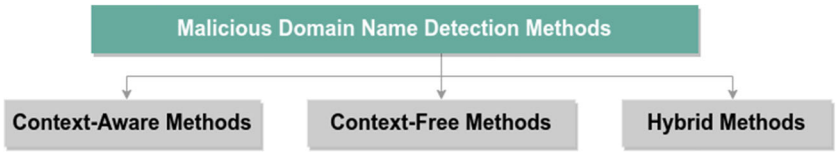

Fig. 1 Taxonomy of malicious domain names detection methods

2. Second, since most studies rely on context-aware characteristics, we began looking for all works that conduct a lexical analysis of domain names rather than a DNS one, excluding all works that combine both.
3. Third, once we had a deeper understanding of the problem through the selected papers, we began to gather papers using a bunch of new keywords to find the most convenient works. The used keywords were inspired by the first collected papers, including typosquatting, DGA, machine learning, deep learning, natural language processing, statistical analysis, textual analysis, linguistic features, and context-free characteristics.
4. Finally, we have selected peer-reviewed articles published in the most trusted and specialized journals. We took into consideration the papers published between 2016 and 2022 that can easily be found in one of the following libraries: Springer, IEEE, ACM Digital Library, Elsevier, and Hindawi. The criteria used are as follows:
    - First, the selected papers use the lexical features of domain names.
    - Second, the papers have to be published recently (between 2016 and 2022).
    - Third, every selected work reports at least one evaluation metric of accuracy, f1-score, recall, or precision that allows us to compare them.

4 Domain names diversion means and techniques

Over the years, attackers have developed several powerful techniques to abuse domain names and their reputations. By doing so, attackers seek to:

- Acquire huge numbers of domain names that can evade existing detection techniques while remaining reliable and relevant to the targeted victims.
- Remedy the lack of IP addresses by associating several domain names with a single IP address.
- Alter lexical characteristics of a domain name very carefully to persuade the users to trust a malicious domain name that looks like the original one without raising the slightest suspicion.

To meet these requirements, attackers develop domain name generation algorithms, domain-flux, and typosquatting techniques to alter a domain name's behavior. Therefore, a deep understanding of such diversion techniques is needed to conceive powerful detection approaches. We explain in the following the operating principle of each alteration technique.

4.1 Domain name generation algorithms (DGA)

Domain name generation algorithms (DGAs) are algorithms that automatically generate a list of domain names based on an initial seed. Cyber-attackers and botnet operators use DGAs to produce hundreds of novel domain names based on time and date. As a result, the deletion of such domain names in zone files gets complicated [5]. Two criteria are considered for the DGAs classification: initial seed and generation scheme.

- **Initial seeding:** these are all the parameters required for the execution of a domain name generation algorithm. The typical parameters include digital constants (for example, the length of the domain name or the seeds for generators of pseudo-random numbers) or characters (for example, alphabet or all possible TLDs). Two seeding properties characterize a DGA:
    - **Time dependence:** which means that the DGA incorporates a source of time for the generation of the domain name.
    - **Determinism:** concerns the availability of parameters, in other terms, whether there is sufficient knowledge of the parameters required for the execution of the DGA that will calculate all possible domain names.
- **Generation scheme:** designates the program's execution logic. There is a total of four widespread options:
    - **DGAs based on arithmetic:** calculating a sequence of values that either has a direct ASCII representation useful for a domain name or designate a lag in one or more hard-coded tables, constituting the DGA alphabet. It is the most common type of DGAs.
    - **DGAs based on the hash:** using the hex digest representation of a hash to produce a DGA. They often use MD5 and SHA256 to generate new domain names.
    - **DGAs based on lists of words:** concatenating a sequence of words from one or more lists of words, which gives fewer random domain names.

– **DGAs based on permutations:** generating all possible items by permutation of an initial domain name.

4.2 Domain-Flux

In order not to figure on a blacklist, botnets must be able to budge in a list of available domain names within a short time. Domain-Flux is a technique that allows bots to obtain this agile behavior. It consists of associating several FQDNs with one IP address [6]. For commandment and control (C&C) attacks, recent botnets such as Conflicker, Kraken, and Torpig use domain re-routing based on the DNS. Each bot calls for a series of domain names. The bot owner must record one sequence for each domain name. This technique makes it difficult unless you have done reverse engineering on the DGA, the call of the so-called root name servers. Authoritative name servers do not generally respond to recursive requests but rather iterative ones, providing the information they have and allowing the applicant to continue his query by following the route provided or blocking the algorithmically generated domain name. These domain names have a very short lifespan.

4.3 Typosquatting

Attackers try, through typosquatting, to record domains by incorrectly spelling the initial domain name, for example, "bnparisbas.com" instead of "bnpparibas.com" [7]. Typosquatting exploits errors made by users when typing domain names in an address bar [6]. The danger behind such a technique lies in its ability to facilitate fraud and lead to the leak of information and corporate secrets [8]. This technique has several variants that try to operate in the same way, namely:

- **Bitsquatting:** that was introduced in July 2011 by Dinaburg [9]. This concept refers to the abusive use of software flipping random bits to generate domain names that have a one-bit difference from authoritative ones [10]. In this case, a random bit-flip in hardware memory where domain names are stored temporarily can lead a user to a malicious domain name, i.e., a domain name that has a character that differs in one bit (such as "micposoft.com") from the same character as the targeted legitimate domain ("microsoft.com")[2]
- **Soundsquatting:** is the practice of registering domain names containing homophone words of authority. When users hit the wrong word and reach the soundsquatted domain name, the soundsquatter, like squatters of generic domains, can then monetize their visit in an illegal way [10].

[2] Example from: https://unit42.paloaltonetworks.com/cybersquatting/.

Example: "whetherportal.com" instead of "wheatherportal.com".

- **Combosquatting:** that designates the attempt to borrow the characteristics of a brand domain name by combining new words with the brand name. The combosquatting does not imply a spelling deviation from the original brand. On the contrary, it requires that the original domain name remains intact [11].
  Example: "apple.com.recover.support"
- **Homographsquatting:** (or homoglyphic attacks) that are committed by substituting characters using glyphs to create deceitful domain names. The latter, however, are hard to distinguish from the real ones [7]. Attackers, in this case, take advantage of internationalized domain names (IDNs), where Unicode characters are authorized to lead such attacks.
  Example: "fäcebook.com"
- **Levelsquatting:** that aims to mislead the victims by putting the brand name in illegitimate subdomains. This attack is particularly deceiving for mobile phone users because the browser address bar may not be wide enough to display the full domain name [12].
  Example: "google.com.virus.com"

5 Malicious domain name detection methods based on context-free characteristics

All the conducted studies in this research area deal with two aspects: data and detection algorithms. Data designs the domain names (collected in large volumes) utilized to infer some properties that help the detection algorithm recognize malicious and legitimate domain names. In this section, we first elaborate on a list of commonly used data sources in the literature, and then we explain every detection technique before synthesizing all the related work to them.

5.1 Data sources

The gap between academia and industry in cybersecurity makes access to data difficult for researchers [13]. Hence, the latter must create their own experimental data sets by performing a preliminary data collection and aggregation phase [14]. Yet, many approaches offer novel domain name properties that enhance detection accuracy. Rather than relying on the same benchmarks, they define their own data sets by adding powerful and impacting characteristics that reinforce detection algorithms.

We established in Table 1, a list including the frequently used domain name sources in the literature. Data used for malicious domain name detection is labeled either malicious or legitimate. Data sources for malicious domain

Table 1 Data sources table

Malicious			Legitimate
Phishing	Spam	DGAs	
PhishTank[3]	MDL[4]	UMUDGA [15]	Alexa Top Sites[5]
OpenPhish[6]	jwSpamSpy[7]	Botnet DGA [16]	The Majestic Million[8]
PhishLabs[9]		360 Netlab[10]	Tranco [17]
		DGArchive[11]	Cisco Umbrella[12]
		AmritaDGA [18]	
		OSINT Bambenek[13]	
		DGARepository[14]	

[3] https://www.phishtank.com/
[4] https://www.malwaredomainlist.com/
[5] https://www.alexa.com/topsites
[6] https://openphish.com/
[7] http://joewein.net/spam/blacklist.htm
[8] https://majestic.com/reports/majestic-million
[9] https://www.phishlabs.com/covid-19-threat-intelligence/
[10] https://data.netlab.360.com/dga/
[11] https://dgarchive.caad.fkie.fraunhofer.de/welcome/
[12] https://umbrella.cisco.com/blog/cisco-umbrella-1-million
[13] https://osint.bambenekconsulting.com/feeds/
[14] https://github.com/andrewaeva/DGA

names include domain names used in phishing, spam, malware, commandment-and-control (C&C), and botnet attacks. Researchers gather data from three domain name blacklists: phishing, spam, and DGAs. Meanwhile, legitimate data sources come from whitelists. These lists objectively evaluate the legitimacy of a domain name based on its popularity and longevity. The list of data sources presented in Table 1 is not exhaustive but contains the most popular data sources that appear recurrently in the literature.

Often, researchers lean on DGAs blacklists as a fundamental source of maliciousness. As a result, DGAs data sources are ultimately the most popular ones in the literature, and 360 Netlab is more particularly the most referenced one in papers. Furthermore, Alexa Top Sites is the most popular source for acquiring benign data samples. Unfortunately, Alexa was retired on May 1, 2022, leaving the place for other alternative solutions. Even though some are dedicated, most are available and open source.

The best example explaining how are these data sources used to create a data set is CIC-Bell-DNS 2021.[3] This data set represents a collaborative project of the Canadian Institute for Cybersecurity with Bell Canada (BC) Cyber Threat Intelligence (CTI). It counts 13011 domain names involved in malware, spam, and phishing attacks, in addition to 500000 benign domain names. It can serve any detection technique when balanced and customized. It provides fourteen impacting lexical features inspired by the literature.

Nevertheless, finding reliable ground truth data is a major challenge for researchers since some data sources are outdated and need to be renovated as soon as possible. In this way, the meticulous collection and integration of new data sources can be followed by the proposal of new effective detection techniques.

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

5.2 Detection methods taxonomy

Nowadays, five categories of methods are used to detect malicious domain names based on their lexical features (Fig. 2). These techniques rely on their ability to exploit a sample of domain names that came from the previously mentioned data sources (Section. 5.1) to predict the nature of new ones, in other terms: before they turn malicious. In this section, we first explain how these methods operate to achieve high-performance results. Next, we give a brief description of every proposed detection system. In the end, we summarize in Tables 2, 3, and 4 all the existing works and report for each one of them, the data sources used for data set creation, the amount of data used for training and preparing the detection method, the algorithms used for detection and the accuracy achieved by the proposed system.

5.2.1 Classical methods

Classical methods measure the statistical difference or the visual similarity between benign and malicious domain names. According to these methods, the similarity or dissimilarity rate between a domain name and a subset of domain names of a similar nature is calculated as follows: if the rate is above a threshold then the novel domain name belongs to the so-called subset otherwise it belongs to the other subset. The

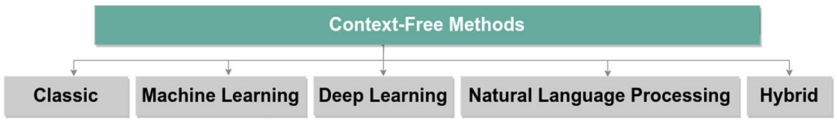

Fig. 2 Taxonomy of context-free methods

imitation characteristics that describe the similarity or dissimilarity rate determine the relationship between benign and malicious domain names [3]. These relationships are generally defined by several metrics [19]. The statistical metrics used in the literature include the following:

- **Divergence of Kullback–Leibler:** that is a non-symmetrical measure of the "distance" between two probability distributions. The divergence (or distance) between two discrete distributions P and Q is given by: $D_{KL}(P||Q) = \sum_{i=1}^{i=n} P(i) \log \frac{P(i)}{Q(i)}$ where n is the number of possible values for a discrete random variable.
- **Jaccard index:** that is defined as $JI = \frac{A \bigcap B}{A \bigcup B}$, where A and B represent all the random variables.
- **Levenshtein's edit distance:** that represents the number of transformations necessary to transform one character chain into another. It is a symmetrical measure that provides intra-domain entropy. The eligible transformations utilized are addition, deletion, and modification.
- **Mahalanobis distance:** that is a distance considering the correlation between the characteristics. The Mahalanobis distance between two vectors of the observations x and γ is calculated as follows: $\sqrt{(x-\gamma)^T C^{-1}(x-\gamma)}$ where C is the covariance matrix of independent variables [20].

Related work

- Authors in [21] proposed the "DomainWatcher" detection system based on three global features: imitation, lexical, and bigram features. They use Levenshtein distance to measure the similarity between known malicious domains and test ones. This work achieves 96% accuracy at best and claims their system to be lighter and faster than state-of-the-art ones.
- Authors in [3] proposed a segmentation approach in which each domain name, except for the TLD, is segmented into substrings. After that, they establish a

Table 2 Summary table of related work

Method	Ref.	Malicious data		Benign/Unlabeled data		Algorithms	Acc.
		Source	Amount	Source	Amount		
Classical	[21]	PhishTank DNS Black Hole	2000	Alexa	5000	Levenshtein distance	96%
	[3]	MDL	2265	Alexa	8000	Reputation threshold	94%
ML	[23]	DGArchive	/	Self-built	/	Random Forest	99%
	[14]	Bader DGAs	32000	Alexa	32000	Random Forest	98%
	[24]	Bambenek	85000	Alexa	85000	Decision Tree	94%
						Ensemble Boosted Tree	94%
						Naive Bayes	92%
						Linear Support Vector Machines	94%
						Coarse K-Nearest Neighbors	94%
	[25]	Bambenek	/	Alexa	/	Support Vector Machines	97.1%
						Logistic Regression	96.6%
						Gradient Boosting	97.6%
						Random Forest	94.7%
						J48 (C4.5)	97%
	[26]	PhishLabs	37610	Self-built	3904	Decision Tree	93%
						Random Forest	96%
						Gradient boosting	97%
						Extreme Gradient Boosting	96%
						Support Vector Machines	96%
						Multiple Layer Perceptron	96%
	[27]	CIC-Bell-DNS	53198	CIC-Bell-DNS	60000	Decision Tree	87%
						Random Forest	88%

Table 3 Summary table of related work (proceeding)

Method	Ref.	Malicious data		Benign data		Algorithms	Acc.
		Source	Amount	Source	Amount		
ML	[28]	360 Netlab	200000	Alexa	100000	Random Forest	90%
	[29]	DGArchive	63 million	Self-built	3.45 million	Random Forest	/[17]
DL	[30]	DGArchive	96000	Alexa	10000	nCBDC	98%
	[31]	Bambenek	765091	Alexa	910313	LSTM_Attention	/[18]
	[32]	[44]	88000	Cisco	352000	HDNN	87.82%
		360 Netlab	100000		400000		97.72%
	[33]	/				LSTM	99%
	[34]	360 NetLab	110000	Alexa	110000	CNN+LSTM_Attention+TCN	99.76%
	[35]	360 NetLab	10000	Alexa	10000	Bayesian LSTM	97%
	[36]	360 NetLab	/	Alexa	/	MHSARNN+SABLSTM	98.9%
NLP	[37]	PhishTank	/	Alexa	/	C4.5	80%
		DNS Black Hole		Self-built			
		MDL					
	[38]	PhishTank	37175	Yandex	36400	Random Forest	97.2%
						Sequential Minimal Optimization	96.4%
						Naive Bayes	75.5%

[17] This work only reports F1-score, recall, and precision
[18] This work only reports F1-score, recall, and precision

substring set, and the weight value of a substring is given by its number of occurrences in the set of substrings. The domain name is segmented using the N-gram method.[4] Its reputation value is calculated based on the weight values of its substrings. Finally, they calculate the threshold that defines the nature of the domain name. The proposed detection algorithm gave an accuracy rate of 94%. They also claim that the time complexity of their algorithm is lower than other popular ones in the literature.

5.2.2 Machine learning (ML) based methods

Machine learning methods are largely used for malicious domain name detection. These methods operate on learning features grouped in one data set by applying further learning algorithms. In the following, we introduce a new taxonomy of learning features and explain the applied learning algorithms to these features before concluding with the related work.

Learning feature Context-free features consider the structural, statistical, and linguistic properties of domain names. Authors in [22] realized a review that detailed several context-free learning features widely used in the literature. In this work, we organize these features rather than enumerate them exhaustively. Ultimately, if more research is conducted in this direction, new features can be proposed. We introduce the following taxonomy for learning features based on our understanding:

- **General features:** where we find information about TLDs, IP addresses, and subdomains that can be found in the domain name sequence intentionally or by mistake. Domain names are supposed to be character sequences that identify an entity. If the so-called character sequence reveals any information related to the network (for instance: IP address) or the entity (for example the company's geographic location), this information is likely to be considered for some detection approaches.
- **Statistical features:** where researchers focus mainly on detecting the randomness in the character sequence by studying its composition and analyzing the characters, digits, and symbols in it. This kind of feature is useful for DGA-based domain name detection since they are deemed to be random and insignificant character sequences. Features like: the domain name length, the length of the consecutive characters, and its number of digits are of high importance in this case.
- **Linguistic features:** where the focus is on the meaning of the domain name sequence rather than its composition. The goal is to identify words alike sequences by segmenting the domains using segmentation methods such as N-grams. This kind of feature analyzes the linguistic value of the domain name. Indeed, it's crucial to exploit information about the meaning if the domain name is intended to mean something.

[4] The process of splitting a domain name into a set of co-occurring character sequences by advancing N characters each time

Table 4 Summary table of related work (proceeding)

Method	Ref.	Malicious data		Benign data		Algorithms	Acc.
		Source	Amount	Source	Amount		
NLP	[39]	360 Netlab	7000	Cisco(2W)	10000	Ensemble(NB,ET,LR)	67.98%
		WB-DGA(2W)	500000	Cisco(2W)	500000		89.91%
		WB-DGA(3W)	50000	Cisco(3W)	50000		91.48%
		WB-DGA(3-4W)	500000	Cisco(3-4W)	500000		80.58%
	[40]	[15]	100000	Cisco	500000	SERM	97.48%
		Cisco	20000		100000		93.80%
		[45–48]	320000		500000		87.60%
		[49]	100000		500000		86.94%
Hybrid	[41]	360 Netlab	200000	Alexa	200000	SBSMW-Convolution+CNN+RF	80%
		DGArchive		Majestic			
	[42]	DGARepository	/	Alexa	/	MLP + Stacking models methodology	99.2%
						SVM + Stacking models methodology	99.3%
	[43]	360 Netlab	337500	Alexa	337500	Random Forest	89%
						Support Vector Machines	96%
						Multiple Layer Perceptron	96%

Learning algorithms Malicious domain name detection using machine learning methods is a classification problem resolved by supervised algorithms or a clustering problem solved by semi-supervised or unsupervised algorithms. These algorithms take as input a vector of attributes or features (Section 5.2.2) that represent the domain name and predict as output the class or cluster of the domain name (malicious or benign). Conventional machine learning algorithms used to solve the malicious domain name detection problem are categorized into three broad categories as follows:

- **Supervised learning algorithms:** that require the labeling of the entire learning set in advance. Each feature vector corresponding to a data sample must be associated with a label representing a class (malicious or benign). These algorithms require huge amounts of labeled and relevant data. However, they seem to be the most appropriate for the problem since they achieve high-performance results. Moreover, the prior use of blacklists and whitelists as classical reactive detection methods helped researchers reduce the data labeling effort. Naive Bayes (NB), Support Vector Machines (SVM), Random Forest (RF), and Decision Tree (DT) are examples of such algorithms that are widely utilized in the literature.
- **Semi-supervised learning algorithms:** that learn from both: labeled and unlabeled data. These algorithms are suitable when the data set is mostly unlabeled but contains some reliable labeled samples. Unlabeled data helps the algorithm improve and prioritize the hypotheses obtained from labeled data. In other words, a semi-supervised classification is done with labeled data samples to acquire some preliminary assumptions. Then, the algorithm is fed with unlabeled data inputs to perform clustering. Cluster-and-label, belief propagation, shortest path, and other graph-based approaches are examples of such algorithms.
- **Unsupervised learning algorithms:** also known as clustering techniques, automatically organize data into groups using unlabeled data sets. Output groups of data are commonly known as clusters. In theory, by carefully selecting features that exhibit different behavior for malicious and benign domains, it is possible to allow clustering algorithms to separate the provided samples into two groups. Then, the researcher decides which group contains which type of domain names. Although these approaches have the advantage of independence from labeled data, they are not very common in the literature. This is mainly due to their design complexity. K-means, K-nearest neighbors, X-means, hierarchical clustering, and fast unfolding are examples of such algorithms.

Related work

- In [23], the authors introduce "FANCI," a Feature-based Automated Non-existent Domain name (NXDomain) Classification Intelligence system. Since a DGA generates a large number of domain names, few of which are registered as valid domain names, many other non-existent domain names are produced. Therefore, the authors analyze the patterns of NXDomains to detect DGA-generated malicious domain names. The proposed system uses the Random Forest classifier. It's

lightweight, generalizable, and usable as-a-service. It achieves 99% of accuracy in the best cases.

- Authors in [14] proposed a machine learning approach using the Random Forest algorithm and relying on purely lexical features of domain names to detect algorithmically generated ones. This approach relies mainly on masked n-grams (vowel/consonant n-grams) but adds general and statistical domain name features. The performance revealed is about 98% accuracy.
- In [24], authors introduce a malicious domain name detection system, "MaldomDetector." The system can detect the domain name before it establishes a successful connection with the server, using only the characters in the domain name. It uses a set of easily computed, language-independent features and a deterministic algorithm to detect malicious domain names. Experimental results show that "MaldomDetector" can maintain high detection accuracy of up to 94% in the best case.
- Authors in [25] propose a machine learning framework including a two-level model. The first model is a supervised classification model where the DGA domain names are classified apart from benign domain names. The second model uses an unsupervised clustering method to identify the algorithms that generate those DGA domain names. The framework achieves an accuracy of 97.6% for the classification model with the Gradient Boosting algorithm.
- Another recent work [26] proposes machine learning-based models using a limited number of features to classify COVID-19-related domain names as either malicious or legitimate. The authors show that a small set of carefully extracted lexical features from domain names can enable the models to achieve high accuracy scores. They add a non-semantic feature: the number of subdomain levels, that impacts predictions. Their best algorithm achieves 96% accuracy in the best case.
- Authors in [27] propose a challenging attack scenario by combining malicious behaviors of Malware, Phishing, Spam, and Botnet with samples of legitimate domains. Two supervised learning algorithms were presented. They obtained an accuracy of 88% using the Random Forest algorithm against 87% for the Decision Tree algorithm.
- In [28], authors proposed a new algorithm for detecting malicious domain names via phishing and DGA-related URLs. First, the statistical characteristics of the URL are extracted into a large data set. Then, they apply a Decision Tree algorithm to classify the obtained data. The test results show that the proposed detection algorithm has an average accuracy rate of 90.31% which means a better performance for detecting malicious domain names regardless of their type.
- Authors in [29] propose the "PUFS" framework to detect DGA-generated malicious domain names by analyzing the patterns of NXDomains. Since NXDomains can be classified into two types: malicious algorithmically-generated domains (mAGDs) generated by DGAs and benign non-existent domains (bNXDs), the training set includes labeled (appearing in DGArchive) mAGDs, unlabeled (not appearing in DGArchive) mAGDs, and unlabeled bNXDs. Therefore, the authors use a positive unlabeled (PU) learning approach that trains a classifier using positive and unlabeled samples. PUFS applies a three-step strategy combining reliable negative (RN) extraction, feature selection, and classifier training. The combination of RN extraction and classifier training achieves PU learning, i.e., learning the patterns of mAGDs from partial labels. The proposed system achieves an F1-score of 99.19%.

5.2.3 Deep learning (DL) based methods

With the rise of deep learning techniques, academia, and industry have begun to present numerous detection methods using deep neural networks since the latter proved their efficiency several times. In the malicious domain name detection field, these methods are also known under the name of featureless models. They are called so because they do not require any attributes and use embedded domain names as inputs. Domain names are converted to ASCII vectors or Unicode values. Therefore, domain name particularities and patterns are inferred by the deep neural network without prior human intervention for feature extraction. These methods propose convolutional, recurrent, or hybrid neural networks to perform the classification. These neural networks are tuned and customized according to the intended purpose.

Related work

- Authors in [30] proposed a new domain classification model by combining characters and n-grams with a deep convolutional neural network (nCBDC). The model does not require manually extracted features. Experiments show that the model reaches a rate of 98% accuracy.
- Authors in [31] propose a DGA domain name classification method based on Long Short-Term Memory with an attention mechanism (LSTM_Attention). They report very interesting results of 95.05% precision, 95.14% recall, and 94.58% F1-score.
- The system proposed by [32] introduces a heterogeneous deep neural network framework (HDNN) for detecting stealthy domain name generation algorithms (SDGA). The proposed HDNN employs an improved parallel

CNN (IPCNN) architecture with a self-attention-based bidirectional long short-term memory (SA-Bi-LSTM) architecture. The system achieves 87.82% accuracy for SDGA domain name detection and 97.72% accuracy on the traditional DGA data set.

- Authors of [33], have proposed an LSTM network for detecting DGA-generated domain names. The main task of the process is to divide the URL into the subdomain, domain, and domain suffix. Then, based on this, the proposed neural network is trained to classify the given train data as malicious or benign. The proposed system performs well with an accuracy level of 99% in the best case.
- In [34], authors proposed a system based on improved deep learning: the combining of three deep neural networks (Convolutional Neural Network (CNN), Temporal Convolutional Network (TCN), and LSTM with attention mechanism (LSTM_Attention)) to obtain a better detection effect than that of the original single or two models. The proposed system achieved a high accuracy rate of 99.76%.
- Authors in [35] introduce a new malicious domain name detection and recognition system. The system starts processing domain names using the character sequence model for feature extraction, and the LSTM with Bayesian Optimization Neural Network for hyperparameter combination optimization, which finally makes the model accuracy above 97%.
- The proposed deep learning approach in [36] uses the Multi-Head Self-Attention-Recurrent Convolution Neural Network-Self Attention Bidirectional Long Short Term Memory model (MHSARNN+SABLSTM) for identifying DGA domain threats. The proposed model achieves 98.9% accuracy and compares its results with other state-of-the-art deep neural networks.

5.2.4 Natural language processing (NLP) based methods

Malicious domain name detection problems can easily be projected on the natural language processing field. Domain names are series of character sequences treated as text. The previously mentioned methods can still be used for classification after NLP mechanisms infer the patterns. However, NLP methods are different in the way they operate. The NLP process takes place as follows:

- **Tokenization:** is the aim of cutting the text into several tokens. The tokens are the simplest elements that can be inferred from a string sequence. This step would indeed be tempting to use a simple cutting into words, that is to say, to separate the words according to the spaces present between them.
- **Syntactic analysis:** makes it possible to identify a representation of the text structure, to highlight the syntactic relationships between words. This step is based on a dictionary (vocabulary) and a set of grammatical rules to determine the syntagms. Syntagms are the sentence constituents. This step is also in charge of organizing the syntagms according to their hierarchy in the sentence.
- **Semantic analysis:** has a double role. It includes two distinct concepts: grammatical and lexical semantics.
  - **Grammatical semantics:** consist in associating a grammatical role with each of the syntagms defined during syntactic analysis.
  - **Lexical semantics:** that is concerned with the meaning of words themselves. We must therefore return to the tokens while considering all the results obtained by the subsequent analysis.
- **Pragmatic analysis:** is the discourse interpretation step. This interpretation can depend on the immediate context or more global knowledge, such as defining a proper dictionary or corpus that brings more information about the context.

Related work

- Authors in [37] proposed a lightweight morpheme feature-based domain name detection algorithm with natural language processing, which analyzed domain name features such as root, affix, Chinese spelling, and special name abbreviation. The algorithm reached 80% of accuracy in the best case.
- In [38], authors proposed a phishing detection system that can detect visual similarities using some natural language processing techniques. They applied some tests on the proposed system. The experimental results have shown good performance with an accuracy rate of 97.2%.
- Authors of [39] exploit the inter-word and inter-domain correlations using semantic analysis approaches, word embedding, and the part-of-speech to detect word-based DGAs. The system achieves an accuracy of 91.48% in the best case using an ensemble classifier constructed from Naive Bayes (NB), Extra-Trees (ET), and Logistic Regression (LR).
- Based on the combining of a collection of semantic elements of domain names (strong, weak, zero), [40] proposes a semantic element representation model (SERM) for domain names. It is constructed based on the analysis of the combinatorial arrangement between elements and Probabilistic Context Free Grammar (PCFG). The DGA

domain names are categorized into four categories: random characters, word-based, predicted characters, and multi-element hybrid. The experimental results show that the SERM achieves an accuracy of 97.48% for random character-based DGA as a best case.

5.2.5 New emerging hybrid methods

A new research direction is emerging in the malicious domain name detection research area. The methods presented and categorized above have recently combined in different manners to solve the problem at hand. Unlike NLP methods, hybridization is not involved in the classification algorithm only but also in all the reasoning behind it. For instance, authors in [41] proposed a malicious domain name detection system that relies on both: deep learning and machine learning techniques depending on the input domain name length. They use an attention-based mechanism to extract features from extra-short domain names and a side-by-side multi-way convolution neural network (SBSMW_Concolution) to perform the classification. For moderate-length domain names, a two-dimensional structure, namely Right Shifted Tensor (RST), is constructed to extract n-gram features besides a convolutional neural network (CNN) for detection. They perform the classification with manually crafted easy-to-calculate features and the Random Forest machine learning algorithm for the extra-long domain names. Then, they conducted their tests on different data sets. They achieved an accuracy of 99% at best. On the other hand, some authors propose combining classical metrics of thresholding with machine learning algorithms of classification to outperform existing DGA domain detection techniques. In [42], authors proposed to consider distance metrics of classical methods as learning features of two machine learning classification algorithms: Support Vector Machines and Multiple Layer Perceptron (MLP) combined with stacked models of Random Forest, XGBoost, LightGBM, and Catboost. A permutation feature importance analysis is presented for explainability. Results show that the proposed system can outperform existing ones, with a detection accuracy of over 99%. Furthermore, authors in [43] present a methodology for detecting algorithmically generated domain names. This approach combines the Kullback-Leibner divergence and Jaccard index metrics as similarity measuring metrics between 2-grams and 3-grams with different machine learning algorithms to classify each domain name as benign or malicious (binary and multi-class classification approaches were conducted. The multi-class classification concerned the DGA family). The proposed methodology achieves good levels of accuracy and leads to a general model capable of efficiently classifying novel domain names.

6 Discussion

As we have seen throughout this paper, researchers are making tremendous progress in the research field of malicious domain name detection. However, the literature lacks an equivalent comparison criterion for comparing existing detection methods. Therefore, we propose to rely on the criterion of targeted-maliciousness type to efficiently compare all the proposed systems. Even though these methods achieve high accuracy rates, they present some weaknesses to be considered while proposing new approaches in order to meet every security use case requirement. Consequently, we list the limits of each detection approach before providing new promising research directions in this research area to be further developed.

6.1 A comparative study of existing work

At this point, we noticed that all existing techniques achieve high-performance results. They are hardly separable if we want to find the most suitable detection method for each type of attack because the secret behind such techniques isn't in the prediction algorithm but in the features and patterns inferred from them. Although the prediction algorithm is a masterpiece in the detection system, the latter is guided, according to the researcher's point of view, by the most impacting characteristics of the domain name. As a result, contributions in this area of research focus on domain name characteristics rather than predictive algorithms. However, comparing these techniques can be based on one relevant criterion neglected for a long time in the literature. We introduce in Table 5 a targeted maliciousness type-oriented comparison of the previously mentioned existing works. We describe the targeted maliciousness type criterion as the attack for which the dataset domain names have been registered. Some datasets target a plethora of attacks while others are more specific. We believe this is the most suitable comparison criterion as there are types of maliciousness that are easier to detect than others due to the nature of the algorithms. For instance, learning algorithms tend to overlearn redundant patterns. Such an algorithm can reveal high accuracy rates when tested on one type of maliciousness only while remaining obsolete when tested on new data such as a new variant of the same attack (for example a new family of DGAs), in which case comparing it to other more varied datasets with low accuracy results is not fair enough. Indeed, DGA-based domain names are easily caught by all the detection systems presented in this literature review, while phishing is much more evasive. Moreover, combining various types of maliciousness definitely decreases the accuracy rate because this combination can bias the classification algorithms, which results in increasing false positive/negative detection rates.

Table 5 Comparative table of related work

Reference	Targeted type of maliciousness	Accuracy
Huang et al. [34]	DGAs	99.76%
Liang et al. [41]	DGAs	99%
Aarthi et al. [33]	DGAs	99%
Wang et al. [42]	DGAs	99%
Schüppen et al. [23]	DGAs	99%
Sarojini and Asha [36]	DGAs	98.9%
Selvi et al. [14]	DGAs	98%
Cucchiarelli et al. [43]	DGAs	98%
Xu et al. [30]	DGAs	98%
GP and Gladston [25]	DGAs	97.6%
Yang et al. [32]	DGAs	97.72%
Yang et al. [40]	DGAs	97.48%
Niu et al. [35]	DGAs	97%
Almashhadani et al. [24]	DGAs	94%
Zhao et al. [3]	DGAs	94%
Yang et al. [39]	DGAs	91.48%
Sun et al. [29]	DGAs	/[19]
Qiao et al. [31]	DGAs	/[20]
Buber et al. [38]	Phishing	97.2%
Zhang et al. [21]	Phishing	96%
Mvula et al. [26]	Phishing[21]	96%
Zhao et al. [28]	Phishing & DGAs	90.31%
Cersosimo and Lara [27]	Malware, Phishing, Spam & Botnets	88%
Zhang et al. [37]	DGAs, Malware & Phishing	80%

[19]This work does not report accuracy
[20]This work does not report accuracy
[21]Specific to COVID-19

6.2 Limitations of state-of-the-art approaches

In this literature review, we synthesized 24 recent works published between 2016 and 2022 all listed in Table 2. These works show to achieve high-performance results. The performance measurement metrics that come up often in the literature are accuracy, F1-score, recall, and precision. Although no method has made less than 80% accuracy, we still found some limitations for these methods, including:

- There is a high computation complexity associated with classical approaches that rely on statistical metrics. Indeed, systems implementing these methods reach their limits very quickly for large numbers of domain names and in a short period since statistical metrics rely on probabilistic variables and exhaustive calculations.
- Machine learning-based methods are generally supervised and require huge amounts of data to achieve high performance. However, data acquisition is getting more complicated because of the privacy policies that preserve the confidentiality of users. Moreover, 70% of existing domain names are of unknown nature, so data labeling is equitably difficult as its acquisition. In addition, results showing 99% accuracy may hide over-fitting. Over-fitting happens when the model learns data by heart and becomes unable to predict for brand-new data. This problem is common to all research fields as it occurs when there is a lack of training data. To detect over-fitting, the learning process must be followed by the validation phase. Furthermore, augmentation mechanisms can be applied to solve the problem if necessary.
- For cybersecurity analysts, fast prediction algorithms are crucial whenever a cyber-attack occurs. The threat response should take place as soon as possible after it is detected, especially when the latter serves a zero-day class attack [14]. Meanwhile, deep learning techniques are slow in terms of training and prediction time. As attacks occur in real-time and predictions must be made in real-time also, deep learning techniques are hindered. These techniques are precise in detection but relinquish in terms of speed/performance trade-off.
- Natural language processing methods do not solve the detection problem effectively because domain names are not always lexical sequences of characters that have a meaning adapted to a single language. On the contrary, they are rarely meant to mean something since domain name holders can opt for abbreviations, brand names, or whatever they want to deploy their resources on the Internet. Moreover, domain names represent brands in several languages. Thanks to Internationalized Domain Names (IDNs), brands can target a specific audience in their mother language without worrying about the relevance of the message they are spreading in other languages.

Therefore, hybridizing detection methods according to some criteria may be a good solution for the limitations stated above. Combining and adapting detection methods according to the security use cases and requirements can help build new up-to-date, reliable, and efficient detection systems.

6.3 Promising research directions

Nowadays, industries are widely deploying malicious domain name detection systems to prevent their digital assets from suspicious attacks. The results obtained by the detection systems presented in this review are easily reproducible in practice. Even if the researchers have made huge progress in this field, we can easily claim that it will remain a promising research area for a long time. Since the artificial intelligence (AI) field is constantly evolving, new research directions are

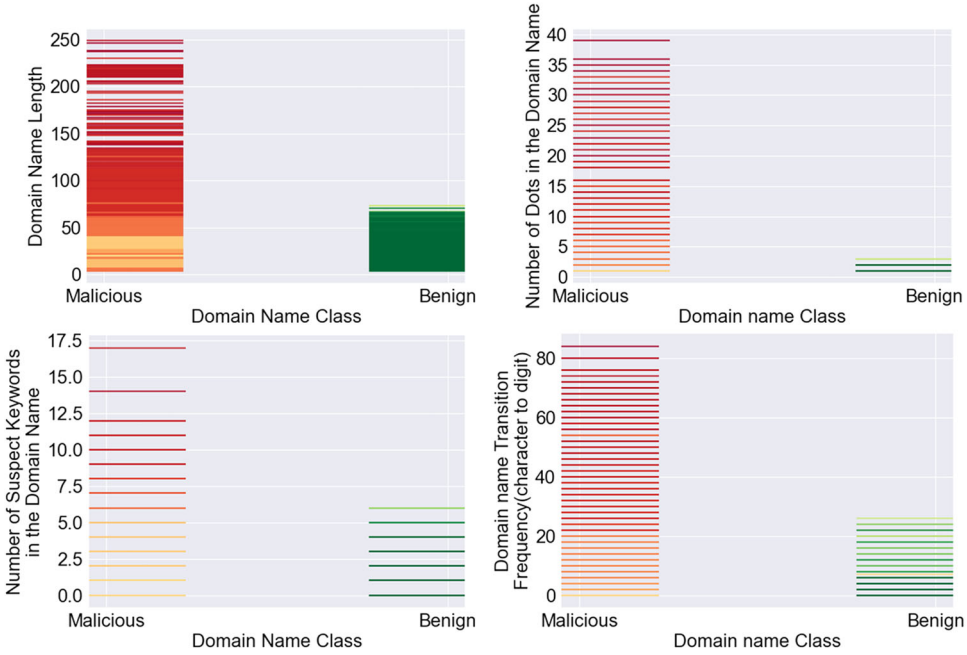

Fig. 3 Characteristics distribution by class

emerging. Some of them standardize new approaches, while others optimize the existing ones.

New algorithms can be tested and customized for malicious domain name detection according to specific needs. Although the Random Forest algorithm proves to be the most efficient in the literature, the proposal of new detection algorithms is a promising research direction. The latter can be tested according to different attack scenarios separately to help researchers find the most suitable algorithms for each type of attack. Furthermore, this will improve the response capability of real-time proactive detection systems to attacks targeting Internet users and domain name holders. Recent literature studies often cover machine learning and deep learning methods since they are innovative and highly customizable.

Machine learning methods have become increasingly popular due to their ability to include domain name-specific attributes. As a result, they achieve high accuracy rates by thoroughly selecting the characteristics that impact the performance metrics most. Thus, attribute selection and extraction can significantly enhance machine learning-based detection systems' performances. We present in Figs. 3 and 4 the distribution of the most popular domain name attributes in the literature by class. In this data visualization process, we explored a customized and balanced data set composed of 4115122 domain names. The data set includes Alexa Top Sites and zone file benign domain names combined with DGARepository (Section 5.1), Phishing.database,[5] and Spam Block List[6] malicious domain names to show the impact of some learning features in distinguishing benign and malicious domain names. We used the Python graphic tool Matplotlib [50] to perform this visualization.

We extracted several learning features from the collected domain names. The features include the domain name length, the number of dots in the domain name, the number of suspected keywords, and the character-to-digit transition frequency. In Fig. 3, we show the distribution of these features according to their classes. Based on this visualization, we can notice that the length of benign domain names does not exceed 75 characters whereas malicious domain names can reach up to 250 characters. Hence, such a characteristic can separate malicious and benign domain names. In addition, benign domain names include very few dots in their composition (not exceeding three dots per domain name), unlike malicious ones including up to 35 dots. This characteristic is specific to levelsquatting domain names that use dots to mimic a sub-domain of a legitimate one while it is intended to attack soon. Additionally, based on the list of suspicious keywords proposed by [7], we visualized the impact they could have on the classification. Malicious domain names contain more suspicious keywords than benign ones. The enrichment and extension of such a list could considerably help machine learning models to improve detection results. Moreover, attributes like: the character/digit transition frequency, the number of digits, and the number of consecutive characters are also popular in the literature for separating malicious and benign domain names. Indeed, low values

[5] https://github.com/mitchellkrogza/Phishing.Database

[6] https://github.com/no-cmyk/Search-Engine-Spam-Blocklist

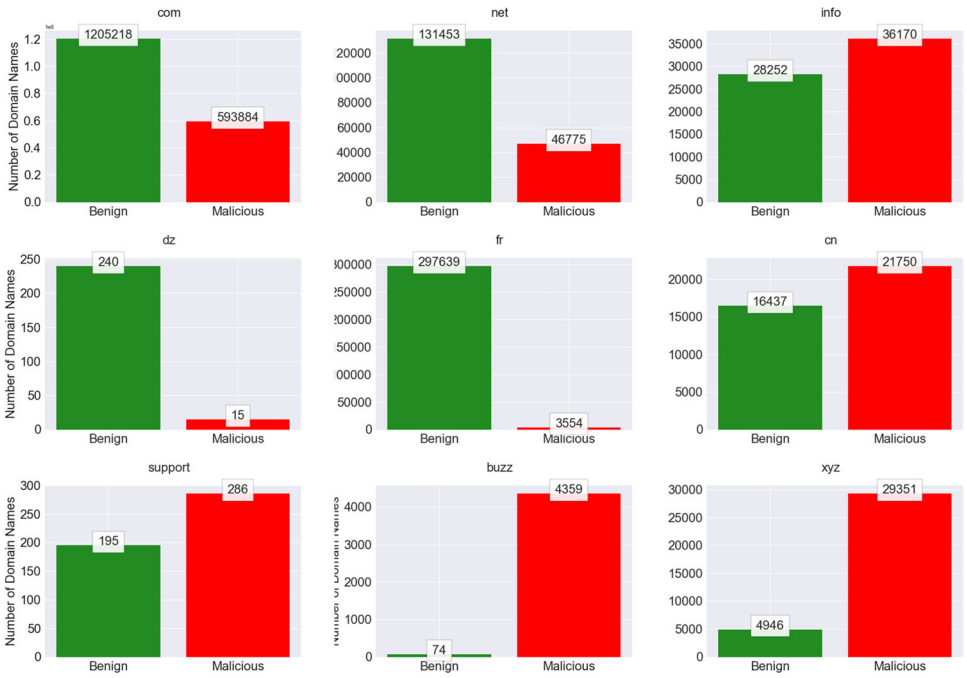

Fig. 4 TLDs distribution by class

of these attributes are commonly found in benign domain names' lexical composition while higher values are found in malicious ones.

In Fig. 4, we show the number of domain names in each TLD according to their classes. The figure shows that some TLDs host more malicious domain names than others. A closer look at these results shows that the ".info" Generic Top Level Domain (gTLD) hosts a higher number of malicious domain names while other gTLDs like ".com" or ".net" include a higher number of benign ones. Also, some Country Code Top Level Domains (ccTLDs) behave similarly. For instance, the ".cn" ccTLD is relatively more exposed to attacks than ".dz" or ".fr" ccTLDs. Moreover, some TLDs are known to host suspicious domain names as seen in Fig. 4 for the TLDs ".support," ".xyz," and ".buzz." Therefore, the TLD attribute can be determined for some detection systems.

Deep learning methods, however, rely on two fundamental factors to enhance their inference capacity and accuracy rates: parameter tuning and attention mechanisms. Parameter tuning is crucial for the learning process since it allows, thanks to manual or automated strategies, to choose the values of the hyper-parameters of the learning algorithms in a targeted way, i.e., if we want to increase the accuracy of the algorithm, we select the parameters that best improve the accuracy by testing different combinations. The attention mechanism is a technique that is meant to mimic human cognitive attention in artificial neural networks by enhancing some parts of the input data and diminishing others. Hence, the focus should be put on these two techniques to optimize deep learning methods.

Regardless of what detection algorithm is used, this one should be able to follow the evolving nature of data and attacks related to domain names. Domain name behavior can change over time following specific topics. Therefore, the suspected keywords that describe malicious topics may become obsolete from one day to another according to geopolitical, financial, or health crises. As shown in Fig. 5, malicious domain names are more likely to follow trends on Twitter than benign ones. Therefore, new data evolution factors should be considered while developing a detection technique.

Furthermore, the literature publications don't fully address maliciousness. Most existing works cover only one malicious type (generally, domain names generated by DGAs). However, the expression "malicious domain names" is vague

Fig. 5 Twitter trending topics distribution according to malicious and benign domain names

Table 6 The distribution of existing works according to the targeted type of maliciousness and the detection method used

	DGAs	Phishing	Spam	Multiple	Total
Classical	1	1	0	0	2
ML	5	1	0	2	8
DL	7	0	0	0	7
NLP	2	1	0	1	4
Hybrid	3	0	0	0	3
Total	17	3	0	3	24

and includes C&C, botnets, phishing, malware, and spam attacks. Upcoming work can target more sources of maliciousness and suggest new powerful algorithms that consider any emerging type of maliciousness by including agile and evolving mechanisms. Table 6 shows the distribution of existing works according to the targeted type of maliciousness and the detection method used. Hence, 87.5% of the proposed detection systems in the literature address only DGA and phishing-related domain names detection. As a result, deep learning-based and hybrid detection methods were never used to detect other malicious sources of domain names. Thus, their potential should be further explored. Furthermore, the table shows that no prior research has addressed spam detection using domain names as an alternative to all content-based and envelope-based spam detection mechanisms. Thus, this research direction is also new and promising.

7 Conclusion

In this paper, we provide a synthesized overview of the research area that deals with malicious domain name detection using context-free features. By placing restrictions on the attributes, precisely targeted fraudulent domain names while respecting Internet users' privacy as context-aware attributes are hardly retrievable and privacy-affecting. To understand how these domain names are created, researchers must begin by developing a clear definition of maliciousness. Therefore, we presented all domain name hijacking means commonly used by attackers to create and register malicious domain names. This way, domain name-based attacks can be countered by thoroughly designed detection methods. These methods deal with the same problem in various ways, whether by threshold calculation like classical methods, learning over-extracted features like machine learning methods, featureless learning like deep learning methods, text analysis like natural language processing methods, or combining several of these methods like hybrid methods. In the discussion, we suggest using a new criterion for comparing detection methods based on the targeted type of maliciousness rather than detection algorithms. We also pinpoint the limitations of every method to help researchers develop techniques that meet the challenging requirements of nowadays. Based on these limitations and requirements, researchers can easily find the most suitable method to use in their detection system. Usually, the goal is to find a satisfactory trade-off between reliability (maximizing the prediction precision), reactivity (minimizing the prediction time), and cost (preserving computational resources). However, there are still a bunch of algorithms to be tested over ground truth data to solve this classification problem. Therefore, researchers must start thinking about gathering more recent data and developing algorithms that can handle data drift and the constantly changing thinking nature of attackers.

Declarations

Competing interests The authors declare no competing interests.

References

1. Hamroun C, Amamou A, Haddadou K, Haroun H, Pujolle G (2022) A review on lexical based malicious domain name detection methods. In: 2022 6th Cyber security in networking conference (CSNet), IEEE, pp 1–7
2. Domain names - implementation and specification. RFC Editor (1987). https://doi.org/10.17487/RFC1035. https://rfc-editor.org/rfc/rfc1035.txt
3. Zhao H, Chang Z, Bao G, Zeng X (2019) Malicious domain names detection algorithm based on n-gram. J. Comp Netw Commun 2019
4. Zago M, Gil Perez M, Martinez Perez G (2020) Scalable detection of botnets based on DGA. Soft Comput 24(8):5517–5537
5. Plohmann D, Yakdan K, Klatt M, Bader J, Gerhards-Padilla E (2016) A comprehensive measurement study of domain generating malware. In: 25th USENIX Security Symposium (USENIX Security 16), USENIX Association, Austin, TX, pp 263–278. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/plohmann
6. Zhauniarovich Y, Khalil I, Yu T, Dacier M (2018) A survey on malicious domains detection through DNS data analysis. ACM Comput Surv 51(4):1–36
7. Fasllija E, Enişer HF, Prünster B (2019) Phish-hook: detecting phishing certificates using certificate transparency logs. In: International conference on security and privacy in communication systems, Springer, pp 320–334
8. Moubayed A, Aqeeli E, Shami A (2021) Detecting DNS typo-squatting using ensemble-based feature selection & classification models. IEEE Can J Electr Comput Eng 44(4):456–466. https://doi.org/10.1109/ICJECE.2021.3072008
9. Dinaburg A (2011) Bitsquatting: DNS hijacking without exploitation. Proceedings of BlackHat Security
10. Nikiforakis N, Van Acker S, Meert W, Desmet L, Piessens F, Joosen W. Bitsquatting: exploiting bit-flips for fun, or profit? In: Proceedings of the 22nd international conference on world wide web.

WWW '13, Association for Computing Machinery, New York, NY, USA, pp 989–998. https://doi.org/10.1145/2488388.2488474
11. Kintis P, Miramirkhani N, Lever C, Chen Y, Romero-Gómez R, Pitropakis N, Nikiforakis N, Antonakakis M (2017) Hiding in plain sight: a longitudinal study of combosquatting abuse. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. CCS '17, Association for Computing Machinery, New York, NY, USA, pp 569–586. https://doi.org/10.1145/3133956.3134002
12. Du K, Yang H, Li Z, Duan H, Hao S, Liu B, Ye Y, Liu M, Su X, Liu G et al (2019) Tl; dr hazard: a comprehensive study of levelsquatting scams. In: International Conference on security and privacy in communication systems, Springer, pp 3–25
13. Rossow C, Dietrich CJ, Grier C, Kreibich C, Paxson V, Pohlmann N, Bos H, Steen MV (2012) Prudent practices for designing malware experiments: status quo and outlook. In: 2012 IEEE Symposium on Security and Privacy, pp 65–79. https://doi.org/10.1109/SP.2012.14
14. Selvi J, Rodriguez RJ, Soria-Olivas E (2019) Detection of algorithmically generated malicious domain names using masked n-grams. Expert Syst Appl 124:156–163
15. Zago M, Perez MG, Perez GM (2020) UMUDGA: a dataset for profiling DGA-based botnet. Computers & Security 92:101719
16. Suryotrisongko H (2020) Botnet DGA dataset. https://doi.org/10.21227/rg6z-z622
17. Le Pochat V, Van Goethem T, Tajalizadehkhoob S, Korczyński M, Joosen W (2019) Tranco: a research-oriented top sites ranking hardened against manipulation. In: Proceedings of the 26th annual network and distributed system security symposium. NDSS 2019. https://doi.org/10.14722/ndss.2019.23386
18. Vinayakumar R, Soman K, Poornachandran P, Alazab M, Thampi S (2019) Amritadga: a comprehensive data set for domain generation algorithms (DGAs) based domain name detection systems and application of deep learning, 455–485
19. Yadav S, Reddy AKK, Reddy ALN, Ranjan S (2010) Detecting algorithmically generated malicious domain names. In: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. IMC '10, Association for Computing Machinery, New York, NY, USA, pp 48–61. https://doi.org/10.1145/1879141.1879148
20. Schiavoni S, Maggi F, Cavallaro L, Zanero S (2014) Phoenix: DGA-based botnet tracking and intelligence. In: International conference on detection of intrusions and malware, and vulnerability assessment, Springer, pp 192–211
21. Zhang P, Liu T, Zhang Y, Ya J, Shi J, Wang Y (2017) Domain watcher: detecting malicious domains based on local and global textual features. Procedia Comput Sci 108:2408–2412
22. Vranken H, Alizadeh H (2022) Detection of DGA-generated domain names with TF-IDF. Electronics 11(3):414
23. Schüppen S, Teubert D, Herrmann P, Meyer U (2018) {FANCI}: feature-based automated {NXDomain} classification and intelligence. In: 27th USENIX Security Symposium (USENIX Security 18), pp 1165–1181
24. Almashhadani AO, Kaiiali M, Carlin D, Sezer S (2020) Maldomdetector: a system for detecting algorithmically generated domain names with machine learning. Computers & Security 93:101787
25. GP A, Gladston A (2020) A machine learning framework for domain generating algorithm based malware detection. Secur Priv 3(6):127
26. Mvula PK, Branco P, Jourdan G-V, Viktor HL (2022) COVID-19 malicious domain names classification. Expert Syst Appl 117553
27. Cersosimo M, Lara A (2022) Detecting malicious domains using the splunk machine learning toolkit. In: NOMS 2022-2022 IEEE/ifip network operations and management symposium, IEEE, pp 1–6
28. Zhao H, Chen Z, Yan R (2022) Malicious domain names detection algorithm based on statistical features of urls. In: 2022 IEEE 25th International conference on computer supported cooperative work in design (CSCWD), IEEE, pp 11–16
29. Sun Y, Jian K, Cui L, Jiang G, Zhang S, Zhang Y, Pei D (2022) Online malicious domain name detection with partial labels for large-scale dependable systems. J Syst Softw 190:111322
30. Xu C, Shen J, Du X (2019) Detection method of domain names generated by DGAs based on semantic representation and deep neural network. Computers & Security 85:77–88
31. Qiao Y, Zhang B, Zhang W, Sangaiah AK, Wu H (2019) DGA domain name classification method based on long short-term memory with attention mechanism. Appl Sci 9(20):4205
32. Yang L, Liu G, Dai Y, Wang J, Zhai J (2020) Detecting stealthy domain generation algorithms using heterogeneous deep neural network framework. IEEE Access 8:82876–82889
33. Aarthi B, Jeenath Shafana N, Flavia J, Chelliah BJ (2022) A hybrid multiclass classifier approach for the detection of malicious domain names using rnn model, 471–482
34. Huang X, Li H, Liu J, Liu F, Wang J, Xie B, Chen B, Zhang Q, Xue T (2022) A malicious domain detection model based on improved deep learning. Comput Intell Neurosci 2022
35. Niu Y, Guan M, Yuan W, Chen Y, Chen L, Yu Q (2022) A Bayesian optimization-based LSTM model for DGA domain name identification approach. In: Journal of Physics: Conference Series, vol. 2303, IOP Publishing, p 012015
36. Sarojini S, Asha S (2022) Detection for domain generation algorithm (DGA) domain botnet based on neural network with multi-head self-attention mechanisms. Int J Syst Assur Eng Manag 1–16
37. Zhang W, Gong J, Liu X, Hu X et al (2016) Lightweight domain name detection algorithm based on morpheme features. J Softw 27(9):2348–2364
38. Buber E, Diri B, Sahingoz OK (2017) NLP based phishing attack detection from URLS. In: International conference on intelligent systems design and applications, Springer, pp 608–618
39. Yang L, Zhai J, Liu W, Ji X, Bai H, Liu G, Dai Y (2019) Detecting word-based algorithmically generated domains using semantic analysis. Symmetry 11(2):176
40. Yang L, Liu G, Wang J, Zhai J, Dai Y (2022) A semantic element representation model for malicious domain name detection. J Inf Secur Appl 66:103148
41. Liang J, Chen S, Wei Z, Zhao S, Zhao W (2022) Hagdetector: heterogeneous DGA domain name detection model. Computers & Security 102803
42. Wang Z, Guo Y, Montgomery D (2022) Machine learning-based algorithmically generated domain detection. Comput Electr Eng 100:107841
43. Cucchiarelli A, Morbidoni C, Spalazzi L, Baldi M (2021) Algorithmically generated malicious domain names detection based on n-grams features. Expert Syst Appl 170:114551
44. Fu Y, Yu L, Hambolu O, Ozcelik I, Husain B, Sun J, Sapra K, Du D, Beasley CT, Brooks RR (2017) Stealthy domain generation algorithms. IEEE Trans Inf Forensics Secur 12(6):1430–1443
45. Fu Y, Yu L, Hambolu O, Ozcelik I, Husain B, Sun J, Sapra K, Du D, Beasley CT, Brooks RR (2017) Stealthy domain generation algorithms. IEEE Trans Inf Forensics Secur 12(6):1430–1443
46. Anderson HS, Woodbridge J, Filar B (2016) Deepdga: adversarially-tuned domain generation and detection. In: Proceedings of the 2016 ACM workshop on artificial intelligence and security, pp 13–21
47. Peck J, Nie C, Sivaguru R, Grumer C, Olumofin F, Yu B, Nascimento A, De Cock M (2019) Charbot: a simple and effective method for evading DGA classifiers. IEEE Access 7:91759–91771

48. Sidi L, Nadler A, Shabtai A (2020) Maskdga: an evasion attack against DGA classifiers and adversarial defenses. IEEE Access 8:161580–161592
49. Yun X, Huang J, Wang Y, Zang T, Zhou Y, Zhang Y (2019) Khaos: an adversarial neural network DGA with high anti-detection ability. IEEE Trans Inf Forensics Secur 15:2225–2240
50. Hunter JD (2007) Matplotlib: a 2d graphics environment. Comput Sci Eng 9(3):90–95