Spotting the Hook: Leveraging Domain Data for Advanced Phishing Detection

Radek Hranický, Adam Horák, Jan Polišenský, Ondřej Ondryáš, Kamil Jeřábek, and Ondřej Ryšavý
Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
Emails: {hranicky, ihorak, ipolisensky}@fit.vut.cz, xondry02@stud.fit.vut.cz, {ijerabek, rysavy}@fit.vut.cz

***Abstract*—Phishing is a major threat, using deceptive tactics to steal sensitive information like passwords and financial details. The rapid innovation by cybercriminals and sophisticated social engineering amplify the challenges in combating phishing campaigns. Traditional blocklisting methods struggle due to the dynamic nature of the Internet and the continuous emergence of new phishing sites. Our research presents an innovative approach to detect phishing domains using machine learning classifiers built upon an extensive array of information combined from DNS records, IP addresses, RDAP servers, TLS certificates, and geolocation data for over 500,000 Internet domains. Using a fine-tailored vector of 143 unique features and seven classification methods, we have achieved a 0.9830 precision rate, an F1 score of 0.9770, and a remarkably low false positive rate of only 0.27%. We further examines the contribution of individual features and the overall impact of information from the utilized data sources on the decision making of the classifiers.**

***Index Terms*—Phishing, Domain, Detection, ML, DNS, IP, RDAP, TLS, GeoIP**

I. Introduction

Phishing is one of the most prevalent malicious threats that users face on the Internet every day [1], [2]. Phishers lurk in the dark, preparing sophisticated campaigns and trying to catch users unaware. Users who are tricked by phishing are exposed to data theft, which often results in significant privacy or financial loss. Attackers typically design phishing sites to mimic legitimate ones, aiming to look trustworthy to the users

The rise in phishing attacks has driven the cybersecurity field to create robust security frameworks. Many research teams and companies propose systems and techniques to combat this menace. Defensive strategies target various levels of user communication, focusing on channels like email, instant messaging, and social networks, where social engineering tactics are quickly deployed. Protection mechanisms aim to identify phishing URLs, domain names, emails, and websites, along with efforts to increase user awareness. Traditionally, phishing protection methods have relied on blocklists and heuristic approaches. Blocklists, while effective, depend on user-reported phishing domains and URLs. However, their scope and frequency of updates are limited, capturing only a fraction of short-lived phishing sites [3]. In recent years, the cybersecurity field has witnessed a paradigm shift with the integration of machine learning techniques. These techniques build on identified data and learn hidden patterns to match similarities, leading to the identification of new threats.

This paper introduces a novel method leveraging machine learning for the real-time detection of phishing domains. We analyze patterns in both benign and phishing domains using a dataset of information for 500,925 domain names, verified and double-checked to ensure the correctness of the ground truth. The information covers data from DNS records, IP addresses, RDAP details, TLS handshakes and certificates, and the GeoIP data. From the dataset, we create a comprehensive feature 143-feature vector on which we trained, tuned, evaluated, and compared seven classifiers. Designed to enhance existing blocklists, our approach adds an advanced layer of defense against emerging phishing threats. It offers a fresh perspective on how data-driven approaches can be used to strengthen digital security in an ever-evolving cyber landscape.

The paper is organized as follows: Section II reviews the evolution of phishing detection techniques. Section III details the data collection methodology. Section IV analyzes the data. Section V discusses feater selection and extraction. Section VI describes the methodology for training and tuning classifiers. Section VII presents experimental results. Section VIII interprets our findings and, finally, Section IX concludes the paper.

II. Related Work

Numerous studies have explored malicious domains, including phishing domains, studying and proposing detection methods. Usable features such as character ratios can be extracted solely from the domain name, as demonstrated by Drichel et al. [4] on 136 lexical features for detecting DGA-based botnet C&C domains.

Bilge et al. [5] highlighted the importance of DNS data in phishing and botnet domain detecting, using two lexical and 15 DNS-related features, including IP address count and TTL properties, obtained from passive DNS traffic analysis. Perdisci et al. [6] similarly employed passive DNS analysis, focusing on statistical IP-related features like IP diversity and average TTL per domain. Antonakakis et al. [7] further confirmed that IP addresses information, such as BGP prefixes, AS numbers, and reputation scores, is highly useful.

An effective phishing detection method is analyzing HTML elements [8], [9]. However, such an approach requires full-page scraping and often rendering, as dynamic content or even single-page applications have become a standard lately. This results in high page-fetching and computational costs. Palaniappan et al. detected malicious domains with DNS and Web-based features [10] using logistic regression. However, their data set consisted of only 20,000 domains, and they reached 60% accuracy on the testing set.

TLS certificate chains provide additional signs of domain maliciousness, as confirmed by Hageman et al. [11] who analyzed phishing attacks in Q4 2020, detecting that 84% of identified attacks were carried out over HTTPS. They also discovered phishers often rely on a small group of issuers, as only 132 of 853 analyzed authorities encountered among certificate chains in phishing campaigns. Torroledo et al. [12] utilized 30 TLS-based features to detect phishing and malware domains, achieving a precision rate of 0.8963. Drichel et al. [13] analyzed certificates from TLS transparency logs, achieving a low false positive rate with 129 features.

Using lexical properties, DNS, or TLS data is contributive, but combining them improves results even further. Kuyama et al. [14] detected malicious domains with 9 WHOIS and 8 DNS-based features, and Shi et al. [15] added two IP-based and three lexical features. Although they showed success, the studies focused primarily on botnet domains. Chatterjee et al. [16] reached a precision of 0.867 in detecting phishing websites with 14 features, including DNS record counts and domain age. However, they focused on URLs rather than domains. Hason et al. [17] detected phishing and C&C domains with 9 features ranked by robustness. Sadique et al. [18] achieved 87% accuracy on a dataset with 38,000 phishing and 60,000 benign domains by merging host-based, WHOIS, GeoIP, and lexical data, the latter having the highest importance. However, no DNS or TLS information was used.

Apart from the study by Sadique et al. [18], most existing ML-based approaches have drawn data from merely one or two sources, for instance, DNS and WHOIS. Moreover, the precision rate of the documented detection methods hardly exceeded 0.9 [12], [16]–[18], indicating a considerably large space for improvements. Previous studies were also generally conducted on smaller datasets, typically ranging between 10,000 and 110,000 samples [13], [15], [16], [18], some had even less samples [14]. Most phishing detection efforts have aimed to identify malicious content on web pages, URLs, or emails. In contrast, methods that examine domain names have been focused primarily on malware C&C domains.

Our work focuses exclusively on phishing detection on a domain-name basis, combining domain lexical features with other available domain-related information from five external data sources. This approach has two notable advantages. Firstly, it allows the detection of phishing in encrypted communication where URLs are not available – in practice, domain names accessed by clients could be collected in a network by observing DNS queries or the SNI fields in most TLS handshakes. Secondly, our method does not require costly scraping, rendering and interpreting the entire page's contents. We propose a comprehensive feature vector consisting of 143 attributes that are used as input to our classifier. Additionally, we crafted a much larger dataset of 500,925 samples to propose and evaluate our classifiers.

III. Data Collection

With machine learning, we faced the challenge of securing ground truth - lists of unquestionably benign and phishing domains. As shown in Figure 1, the first step was building our dataset using publicly available domain lists and conducting additional filtering to eliminate misclassified domains.

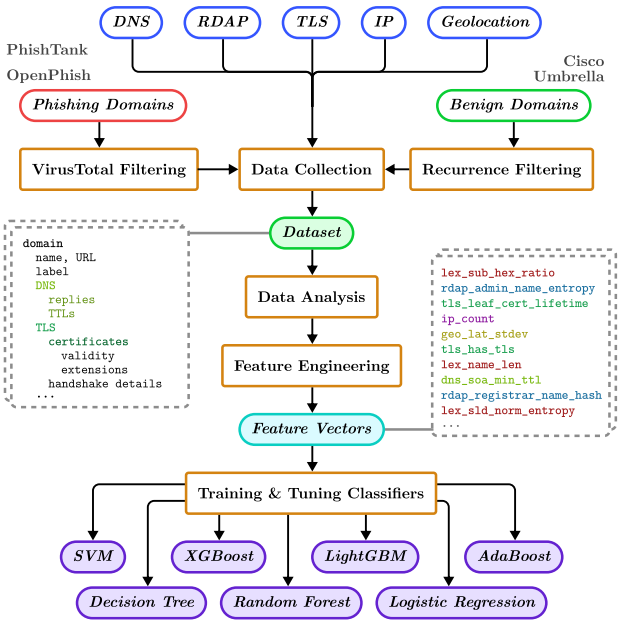

Fig. 1. A holistic overview of the classifier creation.

We chose the public Top One Million list provided by the Cisco Umbrella platform [19] to acquire a set of benign domains for our dataset. The platform was chosen due to its collection methodology, which covers the DNS resolutions of millions of users in more than 150 countries worldwide. It also provides subdomains and is not limited to only domains hosting websites but generally any popular ones regardless of services hosted on the domain, representing a reliable source that mimics a real classifier's input. To ensure only benign domains are in the dataset, we applied recurrence filtering as described by Rahbarinia et al. [20], resulting in a compiled list of 432,572 benign domains.

The phishing dataset was obtained using OpenPhish [21] and PhishTank [22] platforms that accept and validate phishing domain and URL reports. We collected the reports from their MISP feeds as soon as they were published. As our approach is purely domain-centric, we only stored the domain names from the URLs – this resulted in 68,829 potential phishing domains. Stripping the URL data could introduce false positives: for instance, a single phishing resource hosted on a file-sharing service should not necessarily make its entire domain name malicious. To mitigate this issue and increase the quality of the dataset, we conducted additional filtering using VirusTotal [23] that detects malicious URLs based analyses from multiple security vendors. Using the VirusTotal API, 476 misclassified domains were identified and removed. This filtering resulted in a high-quality dataset of 68,353 verified phishing domains.

For each domain name, we performed a DNS scan to obtain information about the records available for the name –

associated IP addresses, name servers, and mail servers. As the domain names were often subdomains of a higher-level zone, we also determined the domain name of the zone by iteratively querying public resolvers for a SOA record, switching to a more general name if no SOA record was found. From the SOA, we determined the primary nameserver address. We then queried this nameserver for the following record types associated with the domain of interest: A, AAAA, CNAME, MX, NS, and TXT. If DNSSEC was present, each RRset signature was validated locally, using only the key retrieved from the domain, without establishing a chain of trust. We also determined the IP addresses corresponding to the names found in the CNAME records.

In the next step, we used all IP addresses obtained from the A, AAAA, and CNAME records. For each IP address, we measured round-trip time (RTT) using ICMP echo and determined their geographic location and autonomous systems affiliation using the publicly available GeoLite2 City and ASN databases (v. 20230606) [24]. Additional information, such as administrative contacts, on DNS zones corresponding to the source names and the related IP addresses was collected using RDAP, or WHOIS if RDAP was unavailable. Finally, for each domain name, we attempted to establish a TLS connection on port 443 to collect certificate chains and handshake details. The final dataset is published on Zenodo[1].

IV. Highlights from the Data Analysis

After collecting all relevant domain-related data, we analyzed and compared benign and phishing datasets. In addition to gaining deeper insight into our data, the motivation was to identify key characteristics that help distinguish between legitimate and phishing domains. This section describes the most interesting findings.

A. Lexical Analysis of Domain Names

The *com* domain, the most common TLD on the Internet [25], appeared most frequently in both datasets. The *net* domain was more common in benign domains, representing 17.5% of the benign dataset, compared to 1.9% in phishing. National TLDs, such as *uk* or *fr*, were much more frequent for benign domains. Domains like *io*, *site*, or *xyz* were frequently used for phishing and very rarely seen in the benign dataset. Phishing sites also often used clickbait TLDs like *page*, *club*, *shop*, *info*, and *online*. Generally, phishing domains had longer names with an average of 29.35 and the median of 26 characters. For benign domains, the average was 22.25, with a median of 20 characters.

Although all domain names mainly consisted of letters, domains with only letters were more prevalent in the benign dataset (56.48%) compared to phishing (32.53%). Figure 2 shows the percentage of individual character groups in domain names. Numbers and hex symbols occurred significantly more in phishing domains. Phishing domain names also had slightly higher entropy.

[1]https://zenodo.org/doi/10.5281/zenodo.12518089

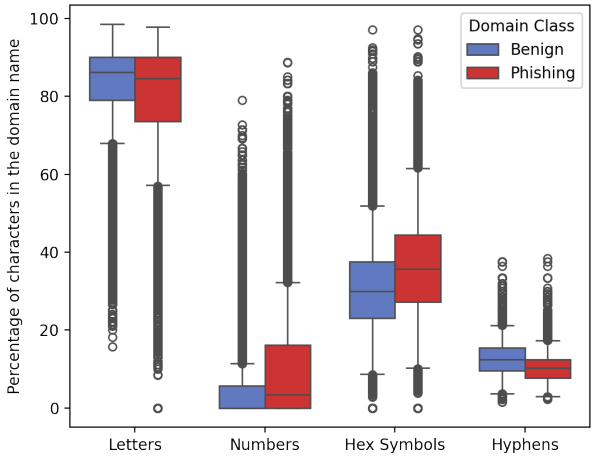

Fig. 2. Character distribution in domain names

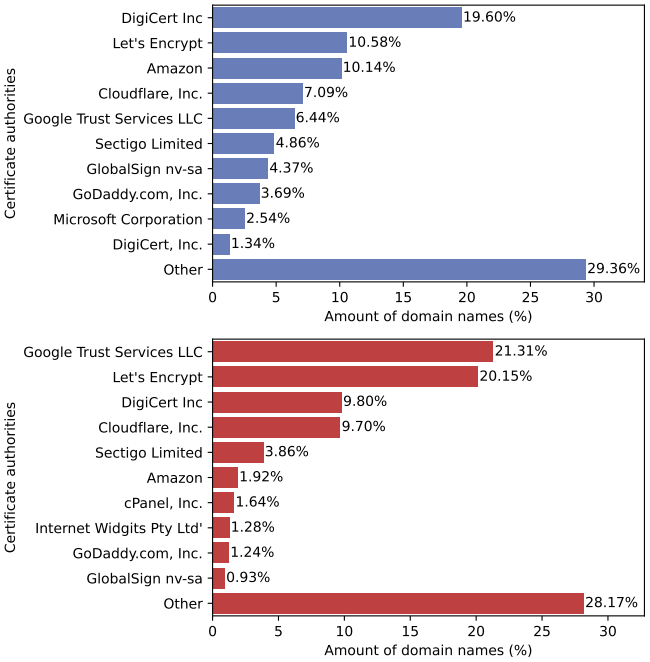

Fig. 3. Top 10 leaf certificate issuers (benign, phishing)

B. Analysis of Data from DNS Records

Inspired by Bilge et al. [5], we examined the TTL values of DNS records. About 0.75% of benign and 0.15% of phishing domain names had very high average TTL values and standard deviations above 80,000. TTL averages and standard deviations showed similar patterns in both sets. Nevertheless, a distinction was observed in domains where 60% to 70% of DNS records had TTL below 100, present in about 20% of phishing domains but only 5% of benign ones.

We also found that phishing domains typically do not have large numbers of DNS records. Less than 1% of phishing domains had more than two A records. In the benign dataset, many domains had large numbers of TXT records, while phishing domains mostly had none, one, or two records.

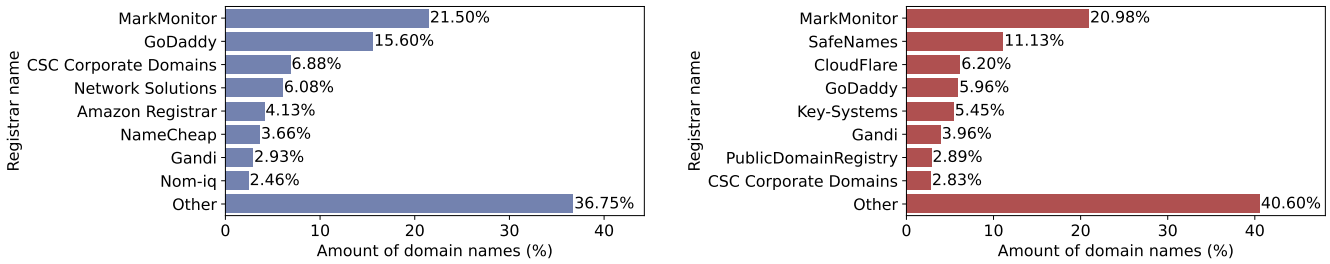

Fig. 4. Top 8 registrars of benign and phishing domain names (registrar names are shortened)

Similarly, a higher number of NS and MX records was more typical of benign domains.

C. Analysis of IP-related Information

The benign dataset showed a higher diversity in the total counts of IP addresses (both IPv4 and IPv6). More than 76% of benign and 85% of phishing domain names have 0 to 3 related IPs. We rarely discovered more than 10 addresses for phishing domains, while some benign domains had over 50, and even over a hundred associated IP addresses.

There are slight differences in IPv6 support. Most of the names in our datasets refer only to IPv4 addresses (73% of the benign names, 65% of the phishing names). In many cases, domains have an equal number of IPv4 and IPv6 addresses (23% of the benign names, 33% of the phishing names). In general, phishing domains tend to have a higher number of IPv6 addresses. IPv6 addresses make up 27% of all IP addresses in the benign dataset, while the number of IPv6 addresses in the phishing dataset is 31%.

D. Analysis of Data from RDAP (WHOIS)

By analyzing domain-related WHOIS/RDAP information, we detected approximately 21% of the domains in both datasets were registered through MarkMonitor, Inc. However, the next top registrars were quite different. The other top two registrars for domains in the benign set were GoDaddy.com, LLC (15.6%) and CSC Corporate Domains, Inc. (6.9%). In the phishing set, SafeNames Ltd. (11.1%) and CloudFlare, Inc. (6.2%). The distribution of the top 8 registrars is shown in Figure 4. Analyzing time properties like the domain age, registration period, and time from last change, we discovered phishing domains are generally short-lived, corresponding to findings of related studies [15], [17], [26].

E. Analysis of TLS Certificate Chains

For domains with available TLS data (77.63% benign, 73.96% phishing), we examined the certificate chains. Only 1.7% of benign and 2.7% of phishing domains used self-signed certificates. Most chains had between two and four certificates: A leaf certificate for the website, 0–2 intermediate certificates, and the root certificate. Having no intermediate authority was common (43.70%) for benign domains, while for phishing it was not that frequent (30.22%). Most phishing domains (59.22%) had chains of length three.

Figure 3 shows the most common leaf certificate issuers. Among benign domains, the most frequent (19.60%) was DigiCert Inc, followed by Let's Encrypt (10.58%) and Amazon (10.14%). For phishing domains, the two dominating leaf certificate authorities were Google Trust Services LCC (21.34%) and Let's Encrypt (20.15%) for obvious reasons. Many Phishing websites were hosted on Google Firebase, where Google allows hosting up to 1 GiB of data at no cost under *web.app* and *firebase.app* domains. Let's Encrypt, on the other hand, provides TLS certificates to essentially anyone who asks, and the process can also be automated easily.

F. Geolocation Data Analysis

Finally, we analyzed the geolocation information for all domain-related IP addresses in both datasets. Unsurprisingly, the highest concentrations of servers were found in the eastern US and western and central Europe. Those included both benign and phishing sites. Nevertheless, some countries, such as Singapore, Taiwan, or Finland, were much more common in the benign dataset, while Brazil, India, Italy, or Vietnam appeared more frequently among phishing locations. In several countries, it was even possible to pinpoint concrete regions where phishing sites were concentrated.

V. Feature Engineering

Following the procedure in Figure 1, the next step was to select features relevant to phishing detection. We started with a list of potentially helpful features. A significant part consists of novel features we designed inspired by our findings from the data analysis. The rest was adopted from previous studies.

This initial selection was filtered in several steps. Specifically, we excluded features that were discussed in related work but could not be extracted from our dataset. Consequently, we removed those deemed irrelevant for phishing detection in previous studies. We also discarded features with no significant contribution and those that duplicated information. Once filtered, we ended with a vector of 143 features that we used for experiments. Table I lists and describes the features, divided into six categories based on their origin. Features with citations are adopted from related work. The rest we consider novel as we have not found studies that cover them.

First, we identified 43 *lexical features* from the domain name alone, as they have proven useful in previous studies [4], [15]. Some features measure the length of the domain name

TABLE I
FEATURE VECTOR FOR PHISHING DOMAIN CLASSIFICATION

Domain Name Lexical Features (lex_)	
Name	**Description & References**
name_len	Length of the domain name [7], [15], [17], [26]
has_digit	Flag if the Domain name (DN) contains a digit [13]
phishing_kw_count	Occurence count of 47 phishing keywords [13]
consecutive_chars	Longest consecutive sequence length [15], [17], [26]
tld_len	Length of the Top-level domain (TLD)
tld_abuse_score	Score for most-abused TLD [27]
tld_hash	Hash of the Top-level domain
sld_len	Length of the Second-level domain (SLD)
sld_norm_entropy	Normalized entropy of SLD
sld_phishing_kw_count	Occurence count of 47 phishing keywords in SLD
sub_count	Number of subdomains (level) [10]
stld_unique_char_cnt	Number of unique characters in TLD and SLD
begins_with_digit	Flag if the name begins with a digit
www_flag	Flag if the name begins with "www"
sub_max_conson_len	Longest consonant sequence length in subdomains [13]
sub_norm_entropy	Norm. entropy of subdomains [4], [15], [18], [26]
{sub,sld}_digit_count	Number of digits in subdomains and SLD [10]
{sub,sld}_digit_ratio	Ratio of digits in subdomains and SLD
{sub,sld}_vowel_count	Number of vowels in subdomains and SLD [18]
{sub,sld}_vowel_ratio	Ratio of vowels in subdomains and SLD
{sub,sld}_consonant_count	Number of consonants in subdomains and SLD
{sub,sld}_consonant_ratio	Ratio of consonants in subdomains and SLD
{sub,sld}_nonalnum_count	Total number of hyphens in subdomains and SLD [10]
{sub,sld}_nonalnum_ratio	Ratio of underscores and hyphens in subdomains and SLD
{sub,sld}_hex_count	Number of hex symbols in subdomains and SLD
{sub,sld}_hex_ratio	Ratio of hex symbols in subdomains and SLD
bigram_matches	No. of common phishing bigram matches [28]
trigram_matches	No. of common phishing trigram matches [28]
tetragram_matches	No. of common phishing tetragram matches [28]
pentagram_matches	No. of common phishing pentagram matches [28]
avg_part_len	Average length of domain name parts
stdev_part_lens	Standard deviation of domain name part lengths
longest_part_len	Length of the longest domain name part
shortest_sub_len	Length of the shortest subdomain

DNS-based Features (dns_)	
Name	**Description & References**
A_count	Number of A records [29]
AAAA_count	Number of AAAA records
MX_count	Number of MX records [14], [30]
NS_count	Number of NS records [14]
TXT_count	Number of TXT records
CNAME_count	Number of CNAME records
resolved_rec_types	Number of discovered RRsets
has_dnskey	Flag if a DNSKEY RRset is in the zone
dnssec_score	DNSSEC scoring
ttl_avg	Avg. of TTLs across RRsets [6], [15], [17], [26], [29]
ttl_stdev	Standard dev. of TTLs across RRsets [15], [17], [26]
ttl_low	Number of RRsets with TTL $\in [0, 100]$ [5]
ttl_mid	Number of RRsets with TTL $\in [101, 500]$ [5]
ttl_distinct_count	Number of distinct TTL values across RRsets [5]
soa_refresh	SOA refresh parameter
soa_retry	SOA retry parameter
soa_expire	SOA expire parameter
soa_min_ttl	SOA minimum TTL
dn_in_mx	Flag if any mailserver is a subdomain of the DN
txt_ext_verif_score	No. of vendor verification strings in TXT RRs
txt_spf_exists	Flag if an SPF record is in the TXT RRs
txt_dkim_exists	Flag if a DKIM record is in the TXT RRs
txt_dmarc_exists	Flag if a DMARC record is in the TXT RRs
DNS-based Lexical Features	
zone_level	No. of subdomains in the zone's DN
zone_digits	No. of digits in the zone's DN
zone_len	No. of characters in the zone's DN
zone_entropy	Normalized entropy of the zone's DN
soa_pri_ns_level	No. of subdomains in the primary NS's DN
soa_pri_ns_digits	No. of digits in the primary NS's DN
soa_pri_ns_len	No. of characters in the primary NS's DN
soa_pri_ns_entropy	Normalized entropy of the primary NS's DN
soa_email_level	No. of subdomains in the admin's mail DN
soa_email_digits	No. of digits in the admin's mail DN
soa_email_len	No. of characters in the admin's mail DN
soa_email_entropy	Normalized entropy of the admin's mail DN
mx_avg_len	Avg. number of characters of the DNs in MX records
mx_avg_entropy	Avg. normalized entropy of the DNs in MX records
txt_avg_entropy	Avg. normalized entropy of TXT RRs values

IP-based Features (ip_)	
Name	**Description & References**
count	Number of IP addresses [5], [7], [15], [17], [26], [31]
mean_average_rtt	Average RTT of all ICMP Echo attempts
ip_v4_ratio	Ratio of IPv4 to all IP addresses
entropy	Total entropy of all /16 (/64 for v6) IP prefixes [6], [32]
as_address_entropy	Entropy of autonomous systems (AS) IP prefixes [32]
asn_entropy	Entropy of AS numbers [10], [18]
distinct_as_count	Number of distinct ASNs [7], [29], [33]

RDAP-based Features (rdap_)	
Name	**Description & References**
Related to the Domain Name	
registration_period	Diff. between expiration and regist. date [15], [17], [26]
domain_age	Days elapsed from the domain registration [29]
time_from_last_change	Days elapsed from the last change [18]
domain_active_time	min(today, expiration) - reg. date [15], [17], [26]
has_dnssec	Flag if domain uses DNSSEC
registrar_name_len	Length of the registrar's name [10], [18], [29]
registrar_name_entropy	Entropy of the registrar's name [10], [18], [29]
registrar_name_hash	Hash of the registrar's name [10], [18], [29]
registrant_name_len	Length of the registrant's name [10], [18]
registrant_name_entropy	Entropy of the registrant's name [10], [18]
admin_name_len	Length of the administrative contact's name
admin_name_entropy	Entropy of the administrative contact's name
admin_email_len	Length of the administrative contact's e-mail [14]
admin_email_entropy	Entropy of the administrative contact's e-mail [14]
Related to Domain-associated IP addresses	
ip_v4_count	No. of IP addresses recognized by RDAP as IPv4
ip_v6_count	No. of IP addresses recognized by RDAP as IPv6
ip_shortest_v4_prefix_len	Length of the shortest IPv4 prefix
ip_longest_v4_prefix_len	Length of the longest IPv4 prefix
ip_shortest_v6_prefix_len	Length of the shortest IPv6 prefix
ip_longest_v6_prefix_len	Length of the longest IPv6 prefix
ip_avg_admin_name_len	Average length of the admin's name for IP addresses
ip_avg_admin_name_ent	Average entropy of the admin's name for IP addresses
ip_avg_admin_email_len	Average length of the admin's e-mail for IP addresses
ip_avg_admin_email_ent	Average entropy of the admin's e-mail for IP address

TLS-based Features (tls_)	
Name	**Description & References**
chain_len	Length of the certificate chain [31]
is_self_signed	Flag if leaf ceriticate is self-signed [12], [31]
root_authority_hash	Hash of root certificate authority's name
leaf_authority_hash	Hash of leaf certificate authority's name
leaf_cert_validity_len	Length of the validity period of the leaf cert. [8], [12], [31]
negotiated_version_id	Negotiated TLS version number (TLSv1.x)
negotiated_cipher_id	An identifier of the negotiated TLS cipher [31], [34]
root_cert_validity_len	Length of the validity period of the root certificate
broken_chain	Flag if there is a certificate that was never valid
expired_chain	Flag if there is an expired certificate in the chain
total_extension_count	Total extensions in all certificates in the chain [12], [34]
critical_extensions	Total extensions flagged as "critical" in all certificates
with_policies_crt_count	No. of certificates that include the *policies* extension
percentage_with_policies	Percentage of certificates with the *policies* extension
x509_anypol_crt_count	No. of certificates not enforcing any policy
iso_pol_crt_count	Total discovered policies from the 1.* OID space
isoitu_pol_crt_count	Total discovered policies the 2.* OID space
subject_count	No. of subject alt. names (SANs) in the leaf cert. [12], [31]
unique_SLD_count	No. of unique domain name SANs
server_auth_crt_count	No. of certs. with "Web Server Authentication"
client_auth_crt_count	No. of certs. with "Web Client Authentication"
CA_certs_in_chain_ratio	Ratio of CA certificates in the chain
common_name_count	No. of common names in the chain

Geolocation Features (geo_)	
Name	**Description & References**
countries_count	Number of distinct countries [5], [7], [15], [17], [26]
countries_hash	Unique hash for each combination of countries [10]
continent_hash	Unique hash for each combination of continents
lat_stdev	Standard deviation from latitudes of IP locations
lon_stdev	Standard deviation from longitudes of IP locations
mean_lat	Mean latitude of IP locations
mean_lon	Mean longitude of IP locations
centroid_lat	Central latitude of IP locations
centroid_lon	Central longitude of IP locations

and its individual subdomain parts. We have also included flags that indicate whether the domain name begins with a digit or with "www". Other important lexical features include character occurrence counts, character ratios, and normalized entropies for the second-level domain (sld_) and for a concatenation of all subdomain parts (sub_). We also calculate the length of the longest consonant sequence. Additionally, we counted the occurrence of 45 common phishing-related clickbait words such as "account" or "free". We also extracted the most common $\{2,3,4,5\}$-grams contained in phishing domains and added features that reflect the number of matches in the analyzed domain name. In addition, we added a feature that reflects the statistical likelihood that the site is abusive based on its TLD. The *tld_abuse_score* ranges from zero to 0.6554, based on data from an article published by Tim Adams [27].

Next, we included 38 *DNS-related features* like counts of record types, as proven useful by Kuyama et al. [14], or records with TTL values in intervals $[0, 100]$ and $[101, 500]$ since Bilge et al. detected that lower TTL values are more frequent for hi-flux malicious domains [5]. We also introduced novel features. To domains that contain a DNSKEY, we assigned a DNSSEC score expressing discrepancies in the signatures, calculated as $(v - 2i)/(v + i)$, where v, i are the counts of valid and invalid signatures respectively. Moreover, we scored the domains by the presence of common verification strings in the TXT records, such as "google-site-verification=". Motivated by the success of lexical features [4], [15], [18]. We also calculated lengths, digit counts, and entropy to various strings found in DNS. In Table I, these attributes are called *DNS-based Lexical Features*.

Eight *IP-related features* describe properties of IP addresses associated with the domain, extracted not only from DNS A and AAAA records but also by resolving IP addresses for CNAMEs. IP address count and IPv4 ratio showed contributions in prior studies [5], [7], [15]. Motivated by the research of Perdisci et al. [6] who suggested low IP diversity often indicates high-flux malicious domains, we included the average entropy of IP prefixes and AS numbers. We suppose credible services may show lower latencies, especially when located in the same area as clients. Therefore, we incorporated the average RTT as a feature. Lastly, we calculated the number of distinct autonomous systems (ASNs).

Next, we included 24 *RDAP-based features*. The first 14 describe domain name registration information like the registration period, domain age, and time from the last change. All are in days and the present timepoint is the midnight of the day when the features were extracted. This way, we ensure the data collection date does not influence the classifiers' decisions. We also added a flag indicating the DNSSEC support. Other features describe the textual properties of the domain's registrar, registrant, and administrative contact. The next ten features are based on the RDAP data collected for the IP addresses. They measure the number of addresses recognized by RDAP as IPv4/6, lengths of prefixes, and textual properties of the IP-related administrative contacts. While we designed most of the RDAP-based features manually, some of them, like the registration period or the domain active time, were inspired by previous studies [15], [17], [26].

Furthermore, 24 *TLS-related features* were extracted from the TLS handshakes and certificate chains. Some features, like the validity length, were adopted from the work of Torolledo et al. [12]. Others, like extensions and security policies, were crafted manually. Features derived from the TLS handshakes describe the negotiated TLS version number and the ID of the negotiated cipher suite. Other TLS-based features reflect the length of the certificate chain, authorities, validity, and various properties based based on extensions and security policies.

Finally, we included nine features related to *geolocation*. The number of different countries has been shown to be a contributing factor in related work [5], [7], [15], [17]. In addition, we computed a unique hash for each combination of country and continent of location, as we believe that specific phishing campaigns may be orchestrated by blackhat groups operating within specific regions. For cases where determining area on a country basis may not be sufficient, we have also included the mean and central latitude and longitude of all the IP locations, as these properties describe concrete areas in a higher level of detail. Standard deviations of latitude and longitude indicate the dispersion of domain-related servers. Such properties help distinguish between services hosted in a smaller region and big national or international players.

VI. Training and Tuning Classifiers

To verify the applicability of our feature selection, we performed the train-test split with 70% of the data reserved for training and tuning and 30% for the final test. We tested the selected features on seven classification methods. For each method, we tuned the model to find the optimal hyperparameter values using a grid search with 5-fold cross-validation [35]. Our main goals were to maximize the F1 score, keep the false positive rate low, and reduce overfitting. Following the described methodology, we tuned and validated the results of the following classification algorithms:

- *Logistic Regression* – The method was chosen as a baseline because it does not rely on linear feature relations.
- *Support Vector Machine* (SVM) – We selected this classifier for its effectivity with high dimensional data and capability of modeling non-linear relationships [36].
- *Decision Tree* – Provides decent performance, clear interpretability of the results, and robustness to outliers.
- *Random Forest* – The method was selected to test how a classifier with many weak learners behaves on our data.
- *AdaBoost* – The method assigns higher weights to relevant features, being beneficial on large feature vectors.
- *XGBoost* – The classifier is known for its high performance and resilience against overfitting [37].
- *LightGBM* – The method was chosen for its effectivity, high training speed, low memory consumption, and native support for categorical features [38].

For the best-performing LightGBM classifier, we utilized 897 estimators of a maximum depth of 17 and 59 leaves. We used a learning rate of 0.15, column subsample ratio of

0.9, `min_child_samples` of 27, and 240,000 samples for constructing bins. The `scale_pos_weight` set to 6.28 compensated the class imbalance.

VII. Experimental Results

First, we verified the performance of the classifiers on the validation part of our dataset. To assess the methods' stability and eliminate the influence of the random state seed choice, we conducted 10 training rounds per classifier with random seeds. Note that the actual effect of randomization depends on concrete configurations of model parameters and is manifested differently in distinct classification methods. As the model may also depend on the order of the samples in the training dataset, we also randomly shuffled the samples in each run.

Table II compares standard metrics' values among the methods when validated on the reserved 30% of the data. For each metric, namely precision, recall, and false positive rate (FPR), the table shows the mean and the variance of all values collected in each round. Due to class imbalance and the goal to eliminate both false positives and false negatives, we consider the F1 score to be the most descriptive metric of success.

We see that the best-performing classifier was LightGBM, which achieved the best results in all metrics. As several related studies use accuracy as the primary metric, we also calculated weighted accuracy, which was 99.39%, with a variance of 7.995e10-9. To get a deeper insight into the classifier's decisions, we examined the impact of each feature. Thus, we used the SHapley Additive exPlanations (SHAP), which also measure the interaction between features [39]. Figure 5 shows the top 20 features according to their SHAP score. The features are sorted by score from most important to least important. Note that the remaining 123 features are not shown, but many of them also contributed significantly to the decision process.

To evaluate the contribution of the different sources of information, we further analyzed how each feature category influences the decision process. Therefore, we calculated the influence I_C of feature class $C \in \{lex, dns, ip, tls, rdap, geo\}$ as an aggregated mean of the absolute SHAP values:

$$I_C = \frac{1}{n} \sum_{i=1}^{n} |\text{SHAP}(f_i)|,$$

where $\text{SHAP}(f_i)$ is the SHAP value for the i-th feature in category C, and n is the number of features in that category. The resulting impact for all categories is displayed in Figure 6. The longer the bar, the more important the category is for the LightGBM classifier.

VIII. Discussion

With an average F1 score of 0.8608, logistic regression fails to grasp complex relations between domain attributes. All other methods achieved a precision above 0.96, F1 above 0.93, and higher resilience to the class imbalance. The Decision Tree classifier produced fair results, but the performance of the Random Forest was naturally higher thanks to the use of hunders of individual trees. SVM and AdaBoost achieved even

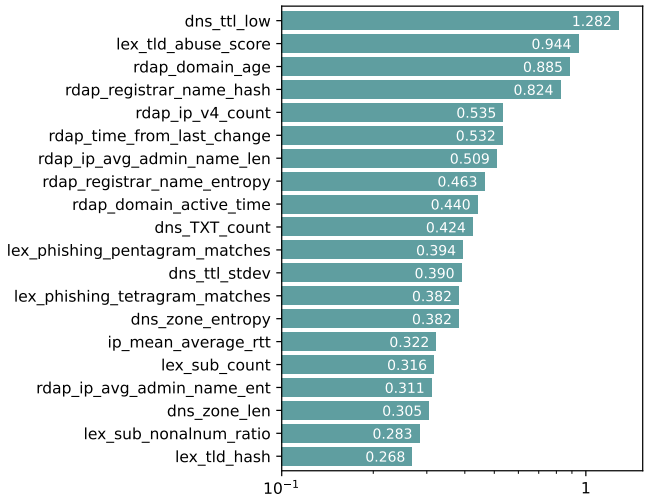

Fig. 5. SHAP score for the 20 most important features (LightGBM)

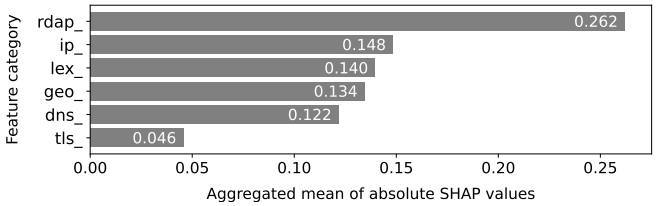

Fig. 6. Impact of individual feature categories (LightGBM)

higher F1 scores, however the methods were highly demanding on computing resources. The best-performing classifiers were XGBoost and LightGBM, with the latter having a slightly higher performance. Both methods were also swift in training. As shown in Table II, LightGBM is a clear winner, outperforming other methods in all metrics. The classifier offers exceptional performance, low training times, resilience to class imbalance, and good handling of categorical features.

The results further show that all the feature categories contributed to the LightGBM classifier decisions, with RDAP having the highest importance. By conducting a separate additional experiment, we verified that classification based solely on RDAP features was less successful, underlying the fact that other sources also play an important role. The most important feature was "dns_ttl_low", confirming the findings of Bilge et al. [5] and our insights from the data analysis. Lexical features also appeared frequently in the top 20 list, namely our proposed statistics-based TLD abuse score and n-gram matching showed usability for building a phishing domain classifier. From RDAP-based features, the most important was domain age, referring to the fact that long-running services are statistically more likely to be trustworthy, while malicious sites are more likely to be shut down early. The information about domain's registrar was also crucial in the decision process, which further confirms its usefulness, documented in previous studies [10], [18]. International service providers such as Facebook or Amazon distribute their servers across

TABLE II
COMPARISON OF RESULTS FOR INDIVIDUAL CLASSIFICATION METHODS

	Precision		**Recall**		**F1**		**FPR**	
Classifier	Avg.	Variance	Avg.	Variance	Avg.	Variance	Avg.	Variance
Logistic Regression (LR)	0.906419	4.00e-08	0.819711	8.24e-08	0.860887	2.92e-08	0.013373	1.06e-09
SVM	0.969702	1.30e-07	0.943646	3.60e-08	0.956541	2.72e-08	0.004659	3.33e-09
DecisionTree (DT)	0.965228	5.73e-08	0.904394	1.76e-08	0.933821	4.75e-09	0.005148	1.39e-09
RandomForest (RF)	0.977666	1.13e-07	0.907915	3.11e-07	0.941500	1.13e-07	0.003277	2.55e-09
AdaBoost (ADAB)	0.970674	5.82e-09	0.957354	1.72e-09	0.963968	1.56e-09	0.004570	1.51e-10
XGBoost (XGB)	0.981501	1.71e-07	0.970540	1.17e-07	0.975990	4.98e-08	0.002890	4.37e-09
LightGBM (LGBM)	**0.983007**	2.11e-07	**0.971004**	4.09e-07	**0.976968**	1.23e-07	**0.002652**	5.39e-09

many nodes around the world, which is a plausible explanation for why the number of IPv4 addresses is so important. The time since the last change of the domain's RDAP records was also important and, as we have additionally verified, contains higher values for benign domains. Higher values indicate that changes were either less frequent or, in the case of no changes, that the domain has existed for a longer period of time, which is also described by the "rdap_domain_age" feature.

Although not included in the top 20 list, geolocation features are an important input for the classifier. The highest SHAP score had the mean latitude, longitude, and hash of countries in which the domain-related IP addresses had servers. Such results confirm our assumptions that certain groups of phishers often operate in different sets of countries. Surprisingly, TLS-based features had much lower impact than other categories, as shown in Figure 6. Nevertheless, the most useful was the negotiated cipher, followed by the certificate chain length.

Compared to existing work, the results look promising. With XGBoost and LightGBM classifiers, we achieved a much lower false positive rate (0.29% and 0.27%) than Bilge et al. [5], who had FPR 1.1% on their dataset. Attempts from Torroledo et al. [12] and Chatterjee et al. [16] showed precision and F1 below 0.90. Hason et al. [17] achieved 0.9292 F1. Our best classifier had 0.9830 average precision and 0.9770 F1 score. Sadique et. al [18] used a method that was closest to our approach and achieved 90.35% Accuracy with Random Forest on batch learning test and 87% Accuracy in a real-time setup. Our best two classifiers both achieved weighted average accuracy over 99%. Note that the datasets differ amongst the studies and many were not available publicly. Moreover, the studies collected different data in diverse ways, making it hardly possible to conduct a direct comparison.

The performance, stability, and adaptability to imbalance make LightGBM robust and reliable for the current problem domain. The proposed classifier, based on a diverse set of data sources, including DNS records, RDAP servers, TLS certificates, IP addresses, and GeoIP data, has proven its applicability in phishing detection. After integrating these diverse data points into a 143-feature vector and further tuning the model, the classifier demonstrates remarkable performance and effectiveness. Thus, our multi-source approach not only demonstrates possible viability in threat-hunting applications, but also provides a nuanced understanding of domain-related attributes essential for identifying phishing activity.

IX. CONCLUSION

We have created a large dataset of domain-related data from which we have identified attributes that are crucial for assessing the credibility of the domain. The applicability of our approach is best demonstrated with boosted ensemble learning methods, which show high effectiveness and low false positive rates. Our results also prove that phishing sites can be detected solely on a domain basis without the need for computationally expensive scraping of web page data.

Furthermore, as the Internet increasingly embraces encrypted communication, it is often infeasible to detect concrete URLs without possessing the TLS encryption keys. However, certificate chains are publicly available, and associated phishing indicators are easy to extract. The same is true for RDAP, DNS, and geolocation information. Our approach can thus be used not only to secure client machines but also in scenarios where phishing activity is detected at the network perimeter. The information about domains contacted is readily available from passive analysis of DNS traffic without the need to decrypt the actual HTTPS sessions. Enriching this data with RDAP, IP, DNS, and geolocation-related information together provides enough valuable clues to detect phishing attempts with a low computational effort and a high success rate.

The results of this study not only advance the understanding of phishing detection techniques but also strengthen the utility of machine learning in cybersecurity. The proposed methodology might be used to deploy classifiers as part of anti-phishing browser extensions, application firewalls, or broader network security systems, contributing to a safer online environment. Nevertheless, practical deployment would require refitting the models over time to match new emerging threats. Moreover, more benign domains that are short-living and specific to the area of deployment shall be used to make classifier withstand the challenges of a real network traffic.

In the future, we intend to test alternative classification methods and implement various optimizations to enhance our classifiers' performance. Most notably, we aim towards deep learning, necessitating modifications to our feature vector. Moreover, we are experimenting with a much larger corpus of data captured directly from data captured directly from an ISP's network. We believe these efforts will improve our phishing detection techniques and introduce more precise decisions, taking the false positive rate to even lower levels.

ACKNOWLEDGMENTS

We would like to thank the OpenPhish Team for granting the permission to use and publish their dataset. We also thank VirusTotal for providing us access to the API for research purposes. The research presented in this paper has been supported by the "Flow-based Encrypted Traffic Analysis" project, no. VJ02010024, granted by the Ministry of the Interior of the Czech Republic and the "Smart Information Technology for a Resilient Society" project, no. FIT-S-23-8209, granted by Brno University of Technology.

REFERENCES

[1] ENISA, *ENISA Threat Landscape 2023*. European Union Agency for Cybersecurity (ENISA), 2023.

[2] M. Mijwil, O. J. Unogwu, Y. Filali, I. Bala, and H. Al-Shahwani, "Exploring the top five evolving threats in cybersecurity: an in-depth overview," *Mesopotamian journal of cybersecurity*, vol. 2023, pp. 57–63, 2023.

[3] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, and A. Doupé, "{PhishTime}: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 379–396.

[4] A. Drichel, N. Faerber, and U. Meyer, "First step towards explainable dga multiclass classification," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–13.

[5] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure: Finding malicious domains using passive DNS analysis," in *NDSS*, 2011, pp. 1–17.

[6] R. Perdisci, I. Corona, and G. Giacinto, "Early detection of malicious flux networks via large-scale passive DNS traffic analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 5, pp. 714–726, 2012.

[7] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou II, and D. Dagon, "Detecting malware domains at the upper DNS hierarchy," in *20th USENIX Security Symposium (USENIX Security 11)*, 2011.

[8] A. Niakanlahiji, B.-T. Chu, and E. Al-Shaer, "Phishmon: A machine learning framework for detecting phishing webpages," in *2018 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 2018, pp. 220–225.

[9] A. Singh and N. Goyal, "A comparison of machine learning attributes for detecting malicious websites," in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*. IEEE, 2019, pp. 352–358.

[10] 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.

[11] K. Hageman, E. Kidmose, R. R. Hansen, and J. M. Pedersen, "Can a TLS certificate be phishy?" in *18th International Conference on Security and Cryptography, SECRYPT 2021*. SCITEPRESS Digital Library, 2021, pp. 38–49.

[12] I. Torroledo, L. D. Camacho, and A. C. Bahnsen, "Hunting malicious TLS certificates with deep neural networks," in *Proceedings of the 11th ACM workshop on Artificial Intelligence and Security*, 2018, pp. 64–73.

[13] A. Drichel, V. Drury, J. von Brandt, and U. Meyer, "Finding phish in a haystack: A pipeline for phishing classification on certificate transparency logs," in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, 2021, pp. 1–12.

[14] M. Kuyama, Y. Kakizaki, and R. Sasaki, "Method for detecting a malicious domain by using whois and dns features," in *3rd international conference on digital security and forensics*, vol. 74, 2016.

[15] Y. Shi, G. Chen, and J. Li, "Malicious domain name detection based on extreme machine learning," *Neural Processing Letters*, vol. 48, pp. 1347–1357, 2018.

[16] M. Chatterjee and A.-S. Namin, "Detecting phishing websites through deep reinforcement learning," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2. IEEE, 2019, pp. 227–232.

[17] N. Hason, A. Dvir, and C. Hajaj, "Robust malicious domain detection," in *Cyber Security Cryptography and Machine Learning: Fourth International Symposium, CSCML 2020, July 2–3, 2020, Proceedings 4*. Springer, 2020, pp. 45–61.

[18] F. Sadique, R. Kaul, S. Badsha, and S. Sengupta, "An automated framework for real-time phishing URL detection," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020, pp. 0335–0341.

[19] Cisco Systems, Inc. (2015) Cisco umbrella. [Online]. Available: {https://umbrella.cisco.com/}

[20] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: Efficient behavior-based tracking of malware-control domains in large isp networks," in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2015, pp. 403–414.

[21] OpenPhish Team. (2014) OpenPhish. [Online]. Available: {https://openphish.com/}

[22] Cisco. (2006) Phishtank. [Online]. Available: {https://phishtank.org/}

[23] Chronicle Cybersecurity. (2012) Virustotal. [Online]. Available: {https://www.virustotal.com/}

[24] MAXMIND. (2002) GeoIP2 Databases. [Online]. Available: {https://www.maxmind.com/en/geoip2-databases}

[25] Tiago Bianchi. (2023, June) Most popular top-level domains worldwide as of June 2023. [Online]. Available: {https://www.statista.com/statistics/265677/number-of-internet-top-level-domains-worldwide/}

[26] C. Hajaj, N. Hason, and A. Dvir, "Less is more: Robust and novel features for malicious domain detection," *Electronics*, vol. 11, no. 6, p. 969, 2022.

[27] Tim Adams. (2020) ScoutDNS Most Abused Top Level Domains List – October 2020. [Online]. Available: {https://www.scoutdns.com/most-abused-top-level-domains-list-october-scoutdns/}

[28] H. Zhao, Z. Chang, G. Bao, and X. Zeng, "Malicious domain names detection algorithm based on n-gram," *Journal of Computer Networks and Communications*, vol. 2019, 2019.

[29] E. Passerini, R. Paleari, L. Martignoni, and D. Bruschi, "Fluxor: Detecting and monitoring fast-flux service networks," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, D. Zamboni, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 186–206.

[30] I. Prieto, E. Magaña, D. Morato, and M. Izal, "Botnet detection based on dns records and active probing." in *Proceedings of 2011 SECRYPT conference*. IEEE, 01 2011, pp. 307–316.

[31] B. Anderson and D. McGrew, "Identifying encrypted malware traffic with contextual flow data," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, ser. AISec '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 35–46. [Online]. Available: https://doi.org/10.1145/2996758.2996768

[32] S. Hao, N. Feamster, and R. Pandrangi, "Monitoring the initial dns behavior of malicious domains," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 269–278. [Online]. Available: https://doi.org/10.1145/2068816.2068842

[33] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for DNS," in *19th USENIX Security Symposium (USENIX Security 10)*, 2010, pp. 273–290.

[34] O. Barut, Y. Luo, T. Zhang, W. Li, and P. Li, "Netml: A challenge for network traffic analytics," 2020.

[35] D. Anguita, A. Ghio, S. Ridella, and D. Sterpi, "K-fold cross validation for error rate estimate in support vector machines," in *DMIN*, 2009, pp. 291–297.

[36] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, September 20-24, 2004. Proceedings 15*. Springer, 2004, pp. 39–50.

[37] K. B. Abou Omar, "Xgboost and lgbm for porto seguro's kaggle challenge: A comparison," *Preprint Semester Project*, 2018.

[38] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[39] L. Merrick and A. Taly, "The explanation game: Explaining machine learning models using shapley values," in *Machine Learning and Knowledge Extraction: 4th IFIP International CD-MAKE Conference, Dublin, Ireland, August 25–28, 2020, Proceedings 4*. Springer, 2020, pp. 17–38.