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, iondryas, ijerabek, rysavy}@fit.vut.cz

***Abstract*—Phishing is a major threat, using deceptive tactics to steal sensitive information such as passwords and financial details. The inventive strategies of cybercriminals coupled with advanced social engineering heighten the difficulties in countering phishing attacks. 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, RDAP servers, TLS certificates, and geolocation data for over 500,000 Internet domains and related IP addresses. Based on a thorough analysis of the data, we propose a fine-tailored vector of 143 unique features that characterize a domain name. We tested the features with seven classification methods and examined their impact on the decision making. The best method achieved a 0.9830 precision rate, an F1 score of 0.9770, and a remarkably low false positive rate of only 0.27%.**

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

I. Introduction

Phishing is one of the most prevalent malicious threats that Internet users face every day [1], [2]. Phishers create sophisticated campaigns to catch users off guard, often leading to data theft, privacy breaches, or financial losses. Phishing sites are designed to mimic legitimate ones, appearing trustworthy to users. The increase in phishing attacks has spurred the development of stronger cybersecurity frameworks. Researchers and companies have proposed systems to combat phishing, focusing on channels like email, instant messaging, and social networks. Protection mechanisms aim to detect phishing URLs, domain names, emails, and websites.

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. They learn hidden patterns in large datasets to match similarities, leading to the identification of new threats.

This paper introduces a novel method that leverages machine learning for real-time phishing detection. We analyze patterns in both benign and malicious domains using a dataset of information about 500,925 domain names, verified and double-checked to ensure the correctness of the ground truth. The information covers DNS records, registration information from RDAP or WHOIS, data from TLS handshakes, certificates, and geolocation information. From the dataset, we created a comprehensive 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.

The paper is organized as follows: Section II reviews the evolution of phishing detection techniques. Section III covers data collection methodology. Section IV analyzes the data. Section V discusses feature selection. 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 detection methods. Usable features like character ratios are extractible solely from the domain name, as demonstrated by Drichel et al. [4] on 136 lexical features for detecting DGA-based botnet domains.

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

An effective phishing detection method is to analyze HTML elements [8], [9]. However, such an approach requires full-page scraping and often rendering, as dynamic content and single-page applications have become a standard lately. This results in high page-fetching and computational costs. Palaniappan et al. [10] detected malicious domains with DNS and Web-based features 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 showed that 84% of identified phishing attacks in Q4 2020 were carried out over HTTPS. They also discovered that phishers often rely on a small group of issuers, as only 132 of 853 analyzed authorities were encountered among certificate

978-3-903176-66-9

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.

Combining features based on the lexical properties, DNS, or TLS data improves the results even further. Kuyama et al. [14] detected malicious domains with 9 WHOIS and 8 DNS-based features, and Shi et al. [15] added 2 IP-based features and 3 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 improvement. Previous studies were often conducted on smaller datasets, typically between 10,000 and 110,000 samples [13], [15], [16], [18]. Most phishing detection efforts have aimed to identify malicious content on web pages, URLs, or emails. In contrast, methods that examine domain names have focused primarily on malware C&C domains.

Following our preliminary research [19], this 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 for 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. 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 an 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 to build our dataset using publicly available domain lists and to perform additional filtering to eliminate misclassified domains.

We chose the public Top One Million list provided by the Cisco Umbrella platform [20] to acquire a set of benign domains. The platform was chosen because of its collection methodology, which covers the DNS resolutions of millions

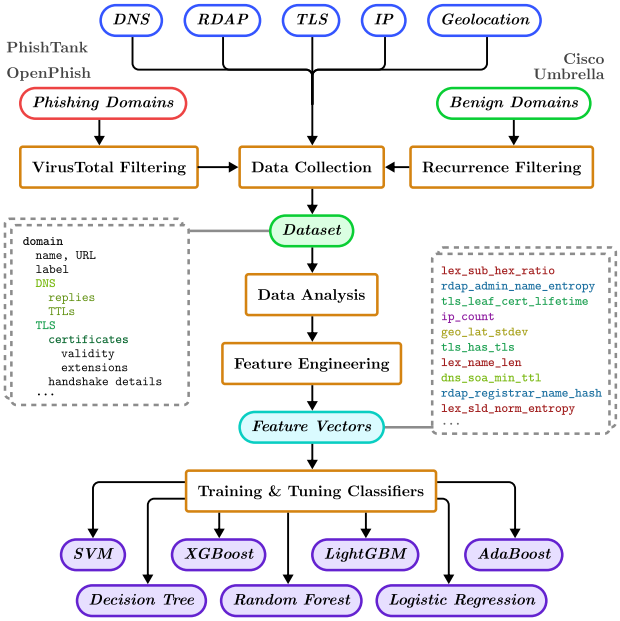

Fig. 1. A holistic overview of the classifier creation.

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. To ensure that only benign domains are in the dataset, we applied recurrence filtering as described by Rahbarinia et al. [21], resulting in a compiled list of 432,572 benign domains.

Phishing domains were sourced from OpenPhish [22] and PhishTank [23], which validate phishing domain and URL reports. We collected reports from their MISP feeds upon publication, storing only domain names. Stripping URL can lead to false positives, e.g., a phishing resource on a file-sharing service may incorrectly make its domain malicious. To address this, we conducted additional filtering with VirusTotal [24], which consults domains and URLs with multiple security vendors. This way, we identified and removed 476 mislabelled domains, resulting in 68,353 verified phishing domains.

For each domain, we performed a DNS scan to gather its available DNS records. As the domain names were often subdomains of a higher-level zone, we also determined the zone domain name. From its SOA record, 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, we locally validated the RRset signatures, without establishing a chain of trust. We also determined the IP addresses for names found in the CNAME records.

For each IP address from A, AAAA, and the resolved CNAME records, we measured the round-trip time (RTT) via ICMP echo from a single specific point and determined their geographic location and affiliation with autonomous systems using the GeoLite2 City and ASN databases (v. 20230606) [25]. We also gathered domain and IP registration data using RDAP (or WHOIS when RDAP was unavailable). Lastly,

we initiated a TLS connection to collect certificate chains and handshake details. Data collection and validation was performed as soon as the domain appeared on a list (received from MISP). The final dataset is published on Zenodo[1].

IV. Highlights from the Data Analysis

After data collection, we analyzed the datasets to identify key characteristics distinguishing legitimate from phishing domains. This section presents some of the notable findings.

a) Lexical Properties: The *net* domain was more common (17.5%) in benign domains, compared to phishing (1.9%). National TLDs, such as *uk* or *fr*, were much more frequent for benign domains. Domains like *io*, *site*, or *xyz* were frequent for phishing but rarely seen in the benign dataset. Phishing sites often used clickbait TLDs like *page*, *club*, *shop*, *info*, and *online*. Phishing domain names tend to be longer (avg. 29.35 characters for phishing, 22.25 for benign). Letter-only domains were more prevalent in the benign dataset (56.48%) compared to phishing (32.53%). Phishing domains also had higher entropy and occurrence of numbers and hex symbols.

b) DNS Data: The mean and standard deviation of the TTL values were similar in both sets. About 20% of the phishing domains had TTL values below 100 in 60% to 70% of their DNS records. Conversely, this applied for only 5% of the benign set. Less than 1% of the 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 0 to 2 TXT records. Higher number of NS and MX records was more typical of benign domains.

c) IP-related Information: The benign dataset showed a higher diversity in the total counts of IPv4 and IPv6 addresses. Over 76% of benign and 85% of phishing domain names had 0 to 3 related IPs. We rarely discovered more than 10 addresses for phishing domains, while some benign domains had over 50. Most of the names in our datasets refer only to IPv4 addresses (73% in the benign and 65% in the phishing set). Domains often refer to an equal number of IPv4 and IPv6 addresses (23% of the benign names, 33% of the phishing names). 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) Registration 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. 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%) were the other top two.

e) TLS Handshakes & Certificate Chains: Having only root and leaf certificate was more common (43.70%) for benign domains, compared to phishing (30.22%). Most phishing domains (59.22%) had chains of length three. Among benign domains, the most frequent leaf certificate issuer was DigiCert Inc (19.60%), followed by Let's Encrypt (10.58%) and Amazon (10.14%). For phishing domains, the dominating leaf certificate authorities were Google Trust Services (21.34%) and Let's Encrypt (20.15%). Many phishing sites 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.

f) Geolocation Data: 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 the phishing sites were concentrated.

V. Feature Engineering

To define features for phishing detection, we started with a list of potentially helpful features based on our findings from the data analysis and previous studies. After initial experiments, we removed those with no significant contribution, and those that duplicate information, resulting in the final vector of 143 features. Table I lists and describes the features, divided into six categories based on their origin. The features with citations are adopted from related work. The rest we consider novel, as we have not found studies that cover them.

We identified 43 *lexical features* based only on the domain name, as the lexical analysis has been proven useful in previous studies [4], [15], [35]. Our features include lengths of the domain and its subdomains, flags for whether the domain starts with a digit or "www", character occurrence and ratios, the longest consonant sequence length, and normalized entropies for the second-level domain (sld_) and for a concatenation of all subdomain parts (sub_). Furthermore, we counted the occurrence of 45 common phishing-related clickbait words such as "account" or "free", and of the most common {2,3,4,5}-grams in phishing domains. 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 published by Tim Adams [27].

We included 38 *DNS-related features* such as record type counts, proven useful by Kuyama et al. [14], or records with TTL values in intervals $[0, 100]$ and $[101, 500]$ since Bilge et al. detected 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=". Inspired by lexical features, we also calculated lengths, digit counts, and entropy for various strings found in the DNS.

Eight *IP-related features* describe properties of IP addresses from DNS A, AAAA records, and resolved CNAMEs. IP address count and IPv4 ratio showed contributions in prior studies [5], [7], [15]. Motivated by Perdisci et al. [6] who suggested that low IP diversity often indicates high-flux malicious domains, we included the average entropy of IP prefixes and AS numbers. As we suppose that credible services may

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

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 (See Section V)
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

show lower latencies, especially when located in the same area as clients, we incorporated the average RTT.

Next, we included 24 *RDAP-based features*. They capture domain registration details, such as registration period or time since the last change, all measured in days with the extraction date as a fixed reference point, ensuring data collection timing does not affect the classifiers. Others describe the textual properties of the domain's registrar, registrant, DNSSEC support, and registration information for the related IP addresses.

Furthermore, 23 *TLS-related features* were extracted from the TLS handshakes and certificate chains. Some, like validity length, were adopted from Torolledo et al. [12], while others, such as extensions and security policies, are novel.

The final nine features relate to *geolocation*. The number of distinct countries showed contributions in previous studies [5], [7], [15], [17]. We also computed unique hashes for combinations of countries and continents, hypothesizing that specific phishing campaigns may originate from particular regions. For finer detail, we added the mean and central latitude and longitude of all IP locations, along with the standard deviations of these to indicate the geographic dispersion of domain-related servers. These features help differentiate between localized services and larger, international operations.

VI. Training and Tuning Classifiers

To verify our approach, we examined seven classification methods using a train-test split with 70% of the data for training and tuning and 30% for final validation. For each method, we tuned the model to find the optimal hyperparameter values using a grid search with 5-fold cross-validation [36]. Our goals were to maximize the F1 score, keep the false positive rate low, and reduce overfitting. Using this methodology, we examined 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 [37].
- *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 [38].
- *LightGBM* – The method was chosen for its effectivity, high training speed, low memory consumption, and native support for categorical features [39].

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

We first evaluated the classifiers' performance on the validation portion of the dataset. To assess stability and minimize the impact of random seed selection, we conducted 10 training rounds per classifier with different random seeds. The effect of randomization depends on model configurations and differs across classification methods. Additionally, since the model's performance can be influenced by the order of training samples, we randomly shuffled the dataset 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.

The best-performing classifier was LightGBM, achieving the highest scores across all metrics. Given that many related studies focus on accuracy, we also calculated the weighted accuracy, which averaged 99.39%. To gain deeper insights into the classifier's decisions, we used SHapley Additive exPlanations (SHAP) to assess feature impact and interactions [40]. Figure 2 displays the top 20 features by SHAP score.

To evaluate the contribution of the different information sources, we analyzed how each feature category influences the decision process. 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 3. The longer the bar, the more important the category is for the LightGBM classifier.

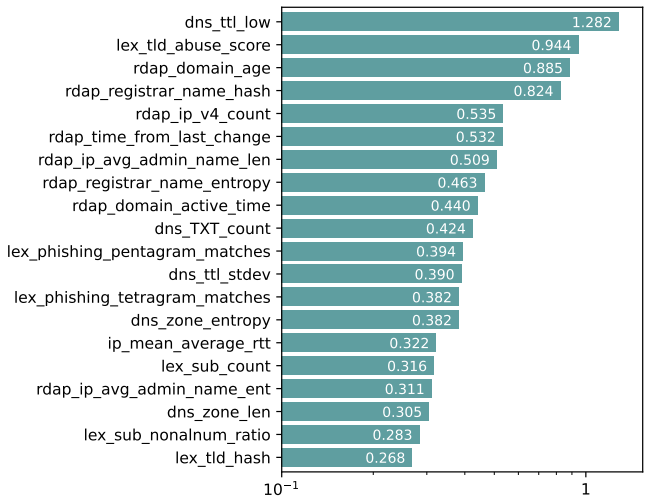

Fig. 2. SHAP score for the 20 most important features (LightGBM)

VIII. Discussion

Logistic regression, with an average F1 score of 0.8608, struggled to capture complex relations between domain at-

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

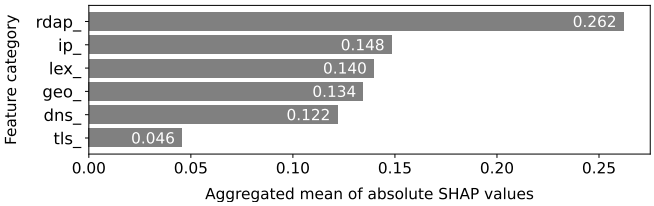

Fig. 3. Impact of individual feature categories (LightGBM)

tributes. In contrast, all other methods achieved precision above 0.96, F1 scores above 0.93, and demonstrated better resilience to class imbalance. While the Decision Tree classifier produced fair results, Random Forest performed better due to the use of multiple individual trees. SVM and AdaBoost achieved even higher F1 scores but were computationally demanding. The top performers were XGBoost and LightGBM, with LightGBM slightly ahead. As shown in Table II, LightGBM excelled in all metrics, offering superior performance, fast training, resilience to class imbalance, and effective 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 appeared frequently in the top 20 list, notably our proposed TLD abuse score and n-gram matching. 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. The 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 many nodes around the world, which is a plausible explanation for why the number of IPv4 addresses is so important. Although not included in the top 20 list, geolocation features are an important input for the classifier. The most contributive geo-based features were mean latitude, longitude, and hash of countries. Surprisingly, TLS-based features had much lower impact than other categories, while the most useful was the negotiated cipher, followed by the length of the certificate chain.

Direct comparisons with related studies are difficult due to significant differences in data collection, processing methodologies, or private datasets. Furthermore, they often focus on entire URLs instead of domain names only. Nevertheless, the results still 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%.

IX. CONCLUSION

We built and published a large dataset of domain-related data to identify key attributes for evaluating domain credibility. Boosted ensemble learning methods proved highly effective, with low false positive rates. Our results also show that phishing sites can be detected solely on a domain basis, without needing full URLs or web page scraping. Publicly available information, such as certificate chains, RDAP, DNS, and geolocation data, provides easy-to-extract phishing indicators, making this approach both practical and efficient. From an ethical standpoint, the dataset contains only publicly available information about services, ensuring that no personal or sensitive data was disclosed.

Our approach can be applied not only to secure client machines but also in detecting phishing activity at the network perimeter. Domain information can be extracted from passive DNS traffic analysis without decrypting HTTPS sessions. By enriching this data with RDAP, DNS, TLS, and geolocation information, we provide sufficient clues to detect phishing attempts with high success. The proposed methodology thus might be used to deploy classifiers as part of anti-phishing browser extensions, application firewalls, or broader network security systems, such as SIEM systems. Practical deployment would require applying adaptive learning techniques, such as refitting the models over time with data from threat intelligence platforms to withstand new emerging threats.

In the future, we intend to test deep learning approaches and implement various optimizations to enhance our classifiers' performance. Moreover, we are experimenting with a much larger corpus of data captured directly from an ISP's network, covering also short-lived benign domains to better match the real traffic. We believe that these efforts will improve our phishing detection techniques and introduce more precise decisions, taking the false positive rate to even lower levels.

ACKNOWLEDGMENTS

We thank the OpenPhish Team for granting permission to use and publish their dataset. We also thank VirusTotal for providing us access to the API for research purposes. This research 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 Tran. 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 *11th COMSNETS Conference*. 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, 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 *43rd Annual COMPSAC Conference*, 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 *10th CCWC Conference*. IEEE, 2020, pp. 0335–0341.

[19] R. Hranický, A. Horák, J. Polišenský, K. Jeřábek, and O. Ryšavý, "Unmasking the Phishermen: Phishing Domain Detection with Machine Learning and Multi-source Intelligence," in *20th Network Operations and Management Symposium (NOMS)*. IEEE, 2024, pp. 1–5.

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

[21] B. Rahbarinia, R. Perdisci, and M. Antonakakis, "Segugio: Efficient behavior-based tracking of malware-control domains in large ISP networks," in *45th Annual ICDSN Conference*. IEEE, 2015, pp. 403–414.

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

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

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

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

[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 *5th DIMVA Conference*. 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.

[32] S. Hao, N. Feamster, and R. Pandrangi, "Monitoring the initial DNS behavior of malicious domains," in *2011 ACM SIGCOMM Conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, p. 269–278.

[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] S. Marchal, "DNS and semantic analysis for phishing detection," PhD thesis, Université de Lorraine, Jun. 2015.

[36] 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.

[37] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *15th ECML Conference, Pisa, Italy*. Springer, 2004, pp. 39–50.

[38] K. B. Abou Omar, "XGBoost and LGBM for Porto Seguro's Kaggle challenge: A comparison," *Preprint Semester Project*, 2018.

[39] 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.

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