BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

http://fit.vut.cz

Project practice 2023/2024

Malicious Domain Detection in the Internet

Tomáš Foltyn*

Abstract

In this technical report, we address the critical challenge of detecting malicious domains on the internet, with a specific focus on phishing and Domain Generation Algorithm (DGA) domains. Leveraging innovative machine learning approaches, our research significantly advances the accuracy and efficiency of detection methods. The report details a thorough exploration of existing literature, the development of a novel methodology, and the execution of experiments that culminate in the creation of DomainRadar—an experimental prototype enhancing threat detection and incident response. Notable contributions include a meticulous dataset analysis and effective hyperparameter tuning, both shaping the research efforts and substantially improving model performance. The report concludes with evidence of significant performance enhancements, high stability, and consistency, underscoring its valuable contribution to the cybersecurity domain.

Keywords: Phishing – Machine Learning – Domain Based Features

Supplementary Material: Downloadable code

*xfolty21@stud.fit.vut.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

In an era dominated by digital connectivity, the internet serves as both a catalyst for progress and a breeding ground for potential threats. The omnipresence of online platforms exposes individuals and organizations to a whole range of risks, from data leaks and identity theft to more insidious cyber threats. The scale of the potential damage is staggering, with implications that go beyond mere financial losses to include critical infrastructure breaches and privacy threats. As we move through this dynamic digital environment, protecting against these threats is an increasingly complex task. Despite continued advances in cybersecurity, the state of detection of malicious activity remains an ongoing challenge that requires innovative approaches to stay ahead of evolving cyber threats. This report sheds light on the ongoing research efforts of our group within the FETA project, in which I have participated as a research assistant to explore innovative approaches to detecting malicious domains on the Internet[1]. The goal of this research is to improve our understanding and strengthen our defenses against evolving cyber threats. The group's scope of research includes the detection of phishing attacks and botnet C&C servers.

Phishing poses a significant cybersecurity challenge, exploiting channels like email and social media to reach unsuspecting users. Despite efforts to educate users on spotting phishing attempts, relying solely on user awareness is insufficient [1, 2, 3]. Even vigilant users can fall victim to sophisticated phishing campaigns, necessitating additional protective measures. Traditional approaches involve using lists or rules, such as allow/deny lists or heuristics, to block known malicious websites [4, 5, 6, 7, 8, 9]. However, these methods may miss zero-hour attacks from new, unreported websites [10]. To address this, researchers turn to machine learning, leveraging various data sources like emails, chats, social media, ads, and domain names [11, 12, 13]. Domain names, crucial identifiers for internet use, play a pivotal role in these efforts [11, 12, 13]. In our group, my focus has been on advancing phishing detection methods, specifically analyzing datasets and improving classifier accuracy.

Domain Generation Algorithms (DGAs) represent a stealthy threat empowering hackers to quickly

[1] More information about the FETA project can be found at https://www.fit.vut.cz/research/project/1530/.en

generate numerous distinct domain names. These adaptable domains serve as fluid pathways for communication, posing challenges in their anticipation and effective blocking [14]. When coupled with the capabilities of botnets — centralized networks of compromised computers — cyber threats gain significant potency. Botnets enable coordinated actions such as the spread of viruses, unauthorized data access, and the execution of distributed denial-of-service (DDoS) attacks [14]. The covert maneuvers of DGAs and botnets pose risks to digital infrastructure, jeopardizing user privacy and sensitive data. Our focus remains steadfast on crafting reliable methods to identify and mitigate the risks associated with DGAs and botnets within the ongoing FETA project, bolstering cybersecurity defenses against these dynamic threats in the process.

2. Relevant networking protocols

The successful detection of malicious activities in the digital landscape relies on a thorough understanding of key networking protocols. In this section, we delve into the Domain Name System (DNS), Transport Layer Security (TLS), and the Registration Data Access Protocol (RDAP). These protocols play pivotal roles in the functioning of the internet, and, as we have discovered, they are crucial components in the development of machine learning-based detection techniques.

2.1 DNS

The Domain Name System (DNS) serves as a fundamental component of the internet, acting as a distributed directory that translates human-readable domain names into machine-readable IP addresses. This systematic translation is vital for navigating the vast network of interconnected devices. When a user inputs a domain name, such as www.example.com, the DNS resolves it by querying a series of servers in hierarchical order until it retrieves the corresponding IP address. Illustration in Figure 1 shows the journey the domain name takes through the different types of servers. This process facilitates seamless communication between devices on the internet.

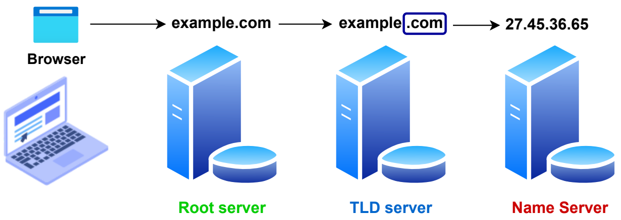

Figure 1. Domain resolution journey: From user query through root, TLD, and name servers to the final IP destination.

Among the frequently encountered DNS records are:

A Record (Address Record): Associates a domain name with an IPv4 address, serving as a primary mapping for most websites.

AAAA Record (IPv6 Address Record): Similar to the A Record but associates a domain name with an IPv6 address, supporting the newer IPv6 protocol.

CNAME Record (Canonical Name): Redirects one domain to another, commonly used for subdomains or aliases.

MX Record (Mail Exchange): Specifies mail servers responsible for receiving email messages on behalf of the domain.

NS Record (Name Server): Identifies authoritative name servers for the domain, essential for delegation and resolving queries.

PTR Record (Pointer): Used for reverse DNS lookups, associating an IP address with a domain name.

SOA Record (Start of Authority): Contains essential information about the domain, including the primary DNS server and contact details.

TXT Record (Text): Accommodates arbitrary text and is often employed for domain verification, DKIM, and SPF records.

In essence, DNS ensures the seamless functioning of the internet by simplifying human interaction with websites through user-friendly domain names while facilitating the behind-the-scenes communication through numerical IP addresses.

2.2 TLS

Transport Layer Security (TLS) is a cryptographic protocol designed to secure communications over a computer network. It operates at the transport layer, ensuring the confidentiality and integrity of data exchanged between two entities, typically a client and a server. TLS succeeds its predecessor, Secure Sockets Layer (SSL), and has become the standard for secure internet communication.

2.2.1 How TLS Works

TLS employs a combination of symmetric and asymmetric encryption to achieve secure communication. When a client initiates a connection to a server, a TLS handshake is initiated. During this handshake, the client and server exchange essential information, including supported cryptographic algorithms and a shared secret key. The process illustrated by Figure 2 involves the following steps:

1. **ClientHello:** The client initiates the connection by sending a message indicating supported cryptographic algorithms and other parameters.
2. **ServerHello:** The server responds by selecting the most robust encryption algorithm from the client's list and communicating it back to the client.
3. **Key Exchange:** The client and server exchange information to generate a shared secret key without exposing it to potential eavesdroppers.
4. **Finished:** Both parties confirm the completion of the handshake, indicating that subsequent communication will be encrypted.

Once the handshake is complete, the actual data exchange occurs using the agreed-upon encryption method, providing confidentiality and integrity.

2.2.2 Use Cases of TLS

TLS is widely used to secure various online activities, including:

- **Web Browsing:** TLS secures the data transmitted between web browsers and servers, ensuring that sensitive information such as login credentials and personal data remain confidential.
- **Email Communication:** TLS safeguards email communication, preventing unauthorized access to email content during transmission.
- **File Transfer:** TLS is commonly employed to secure file transfers, especially in protocols like HTTPS and FTPS.

2.2.3 Comparison with SSL

TLS and SSL share the same objective of securing communications, but TLS is an improved version with enhanced security features. One significant difference is the cryptographic algorithms used. TLS employs stronger and more secure algorithms, making it less susceptible to vulnerabilities that plagued SSL. Additionally, TLS supports forward secrecy, a crucial feature ensuring that even if the long-term secret key is compromised, past communications remain secure.

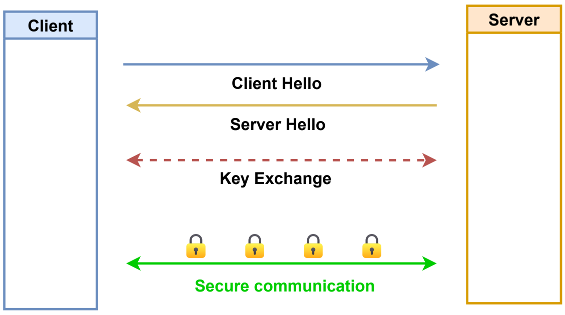

Figure 2. Visualizing TLS Handshake: Secure initiation, key exchange, and encrypted data flow.

2.3 WHOIS/RDAP

The Registration Data Access Protocol (RDAP) is a standardized protocol designed to provide access to domain registration information. It serves as a successor to the traditional WHOIS protocol, offering a more structured and extensible approach to accessing domain registration data.

2.3.1 Overview of RDAP

RDAP was developed to address the limitations of WHOIS, offering a more consistent and machine-readable format for querying domain registration information. Unlike WHOIS (based on TCP/UDP), RDAP operates over HTTP, making it compatible with modern web technologies and allowing for easier integration into applications and services.

2.3.2 Information Obtainable with RDAP

In the context of the RDAP, a comprehensive set of information can be obtained, including:

Entity Details: Information about the registered entity, such as the domain holder's name, contact details, and organizational information.

Registration Status: Details regarding the current status of the domain registration, including expiration date and renewal information.

Name Servers: Information about the authoritative name servers associated with the domain.

Registration Events: Historical data on registration events, enabling tracking of changes over time.

2.3.3 Use Cases of RDAP

RDAP finds application in various contexts, such as:

Cybersecurity: RDAP is crucial for cybersecurity professionals to investigate potentially malicious domains, aiding in the identification of threat actors and their infrastructure.

Law Enforcement: Authorities use RDAP to obtain accurate and up-to-date information for legal and investigative purposes.

Domain Management: Registrars and domain administrators leverage RDAP to manage and monitor domain portfolios effectively.

2.3.4 Comparison with WHOIS

Compared to its predecessor, RDAP offers several advantages:

Standardization: RDAP follows a standardized data format, enhancing interoperability and ensuring consistent data representation across different registries.

Authorization and Authentication: RDAP supports authentication mechanisms, allowing for controlled access to sensitive registration data and addressing privacy concerns.

Internationalization: RDAP supports international characters, accommodating domain names in various languages and scripts.

Extensibility: RDAP allows for extensions, enabling registries to include additional information beyond the core data elements.

In summary, RDAP represents a modernized and improved approach to accessing domain registration information, offering enhanced functionality and addressing the shortcomings of the traditional WHOIS protocol.

3. Machine learning techniques for detecting harmful domains

In the ever-evolving landscape of cybersecurity, combating threats such as phishing, malware, and Command & Control (C&C) Domain Generation Algorithms (DGA) has become a paramount concern. Traditional countermeasures exhibit limitations, prompting a shift towards machine learning (ML) as a robust defense. This section explores key principles and approaches in utilizing ML for detecting malicious domains on the Internet.

3.1 Phishing Detection

Phishing, a prevalent cyber threat, necessitates sophisticated detection mechanisms. Machine learning models have proven effective in discerning phishing emails, domains, and URLs. Noteworthy approaches include:

- **Structural Analysis:** Chandrasekaran et al. employed Support Vector Machines (SVM) with 25 features, achieving a 95% detection rate by incorporating keyword searches for terms like "account" or "security" [11].
- **PILFER Algorithm:** Fette et al. introduced the PILFER algorithm, utilizing Random Forest to achieve a 96% accuracy in detecting both phishing emails and websites [12].
- **Comparative Analysis:** Abu-Nimeh et al. conducted a comparative study, exploring multiple classification methods with precision reaching up to 95.11% [13].

3.2 Malicious Domain Detection

The detection of malicious domains in general involves leveraging lexical, DNS, and TLS-based features. Diverse studies contribute to this multifaceted approach:

- **Lexical Features:** Drichel et al. validated the effectiveness of examining 136 lexical features for identifying Command & Control (C&C) domains, emphasizing the importance of domain name attributes [15].
- **DNS Analysis:** Bilge et al. identified phishing and botnet domains through passive DNS traffic analysis, incorporating features such as the count of distinct IP addresses and countries [16].
- **TLS Certificates:** Hageman et al. highlighted the significance of TLS certificates, reporting that 84% of detected phishing attacks occurred over HTTPS, emphasizing the relevance of TLS-based features [17].

3.3 Integration and Advancements

To enhance detection accuracy, integrating features from multiple sources has emerged as a promising strategy. Kuyama et al. successfully combined lexical, WHOIS, and DNS-based features for detecting malicious domains [18]. However, most ML-driven techniques have room for improvement, with many struggling to surpass 90% precision.

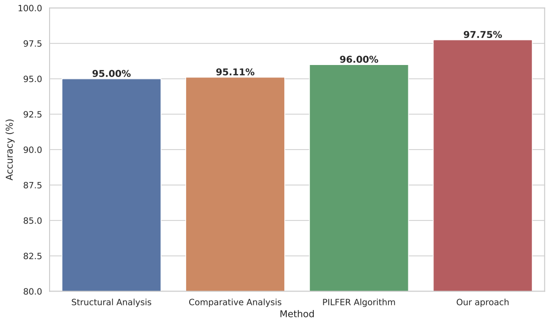

Figure 3. Comparison of accuracy reached via various ML methods. Our latest results included.

While progress has been made, challenges persist. Many existing techniques focus on web page contents, URLs, or emails, leaving domain-centric analyses less explored. Acknowledging these challenges, our research aims to combine lexical domain name attributes with data from five external sources, striving for a comprehensive and effective approach. The results of our latest advancements will be discussed in Section 6. The comparison between our approach and the other mentioned methods is illustrated in Figure 3.

4. Ongoing Research and Innovations

In the pursuit of fortifying cybersecurity measures, the Project FETA – QRadar Group has made significant strides in various domains, focusing on cutting-edge research in phishing detection, DGA detection, and the development of DomainRadar—an experimental prototype for threat detection and incident response.

4.1 Phishing Detection

Motivated by the limitations of small datasets in existing research, our team delved into unmasking phishing activities through multi-source intelligence. In the upcoming IEEE NOMS 2024, our research, titled *Unmasking the Phishermen*, explores detection techniques based on various data sources such as lexical, IP, ICMP, DNS, TLS, RDAP, and GeoIP. Notable findings include insights into certification authorities, DNS TTL intervals for malicious domains, and the crucial role of lexical information in both DGA and phishing detection. For preliminary performance results of the data configuration, XGBoot Random forest classifier was used. Figure 4 depicts a Confusion matrix for the Phishing domain classifier, providing a visual representation of the classifier's performance.

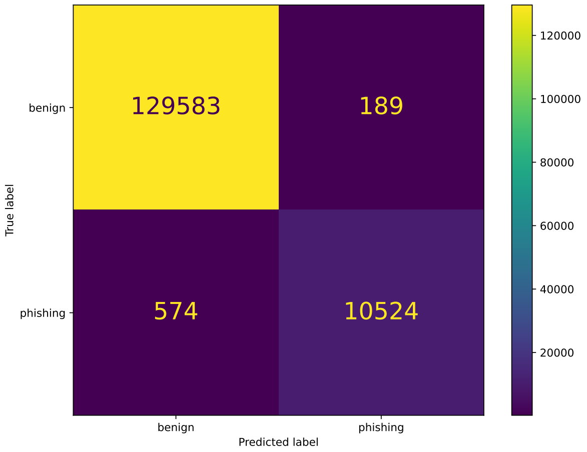

Figure 4. Confusion matrix for the Phishing domain classifier

Scheduled for ACM ASIACCS 2024, our second article, *Spotting the Hook*, conducts in-depth analysis with a broad feature vector (128+ features) and compares seven classification methods. These efforts aim to advance phishing detection methods and contribute valuable insights to the academic community.

4.2 DGA Detections

The focus on detecting Domain Generation Algorithm (DGA) domains has yielded impressive results. Leveraging binary and multi-class classifiers, the team has achieved precision, recall, F1 score, and accuracy percentages that speak volumes about the efficacy of the algorithms.

Classifier	Precision	Recall	F1 Score	Accuracy
Binary	98.4%	97.8%	94.22%	98.59%
Multi-class	87.3%	87.8%	86.2%	87.8%

Table 1. DGA Detection Performance Metrics

Table 1 encapsulates the precision, recall, F1 score, and accuracy achieved by the binary and multi-class classifiers. These metrics emphasize the robustness of the DGA detection system, showcasing its high accuracy and reliability.

4.3 DomainRadar

DomainRadar, an advanced threat detection and incident response application, employs machine learning techniques, specifically Support Vector Machines (SVM) and Neural Networks (NN), to unveil communication with potentially malicious domains within a network [19]. The application's capabilities are outlined as follows:

- **Threat Assessment:** Each domain is assigned a Badness ratio, providing a clear indication of its potential malicious nature. This aids in prioritizing and responding to potential threats efficiently.
- **Classification Sources:** DomainRadar draws on diverse data sources for classification, including domain name syntax, geolocation data, WHOIS data, TLS/SSL certificate information, and DNS data. This multi-faceted approach enhances the accuracy and comprehensiveness of domain classification.
- **Integration with Other Systems:** DomainRadar seamlessly cooperates with Suricata IDS and QRadar SIEM. It extracts and analyzes data from QRadar SIEM, encompassing offenses and DNS traffic, to provide a holistic view of potential security incidents.
- **Backend Architecture:** The backend architecture consists of essential components, including the Flask App, Loader, Resolver, and Analyzer daemons. These work in tandem to facilitate data loading, external data resolution, and domain analysis.
- **Incremental Processes:** To ensure efficiency, DomainRadar adopts incremental processes for data

loading and domain evaluation. Only new or updated information is incorporated into the system, optimizing resources and preserving historical data.

Overall, the research group made substantial contributions to the domain of identifying malicious domains. Following these contributions, the group transitioned towards applying their findings in practical scenarios.

5. My contribution

In this section, I describe my unique work for the project, emphasizing two key aspects: detailed data analysis and the adjustment of hyperparameters for classification models.

5.1 Dataset analysis

My main goal was to study the data collected. The current data analysis, although helpful, had some problems. To overcome these problems, I did a more thorough analysis, covering different aspects of the data, including lexical, IP, RDAP, DNS, and geolocation attributes.

5.1.1 Lexical featuers

The lexical-based features include the examination of the top-level domain (Fig. 6 and 5), domain name length and entropy (Fig. 7), shortest and longest part length, average part length (Fig. 8), and subdomain count. The distribution of characters was also examined (Fig. 9). These features provide insights into the structural composition of domain names, aiding in understanding naming conventions and potential patterns.

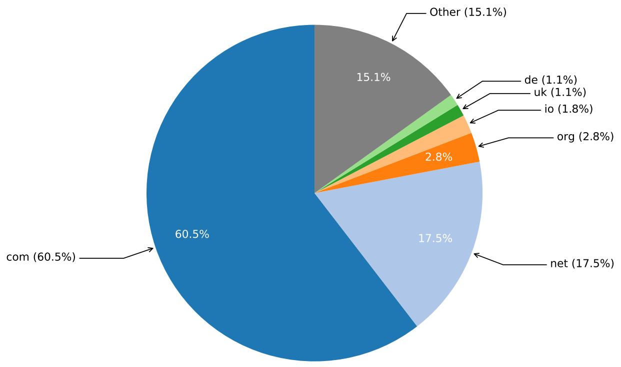

Figure 5. Top-level domains of the Benign dataset

5.1.2 IP and RDAP based

The IP-based features focus on the percentage of IPv4 addresses per domain and their distribution across the entire dataset (Fig. 10). This analysis provides a glimpse into the prevalence of IPv4 addresses in different domains, contributing to the broader understanding of address allocation trends. The RDAP-based features involve the examination of registrar names (Figs. 12 and 11) and the distribution of entity types, such as administrative, registrant, registrar, abuse, admin, and technical entities (Fig. 13). These features offer valuable information about the organizational and administrative aspects associated with domain registration. Additionally, domain age (Fig. 14) is considered as an essential RDAP-based feature, providing temporal insights into the lifecycle of domains.

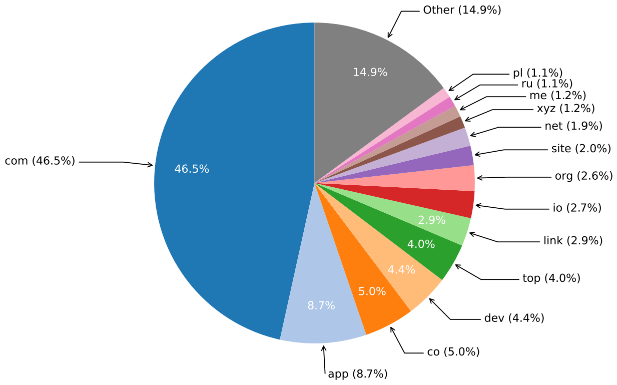

Figure 6. Top-level domains of the Phishing dataset

5.1.3 DNS based

The DNS-based features encompass TTL statistics, including average, standard deviation, low, mid (Fig. 15), and distinct counts (Fig. 16). These metrics offer a nuanced perspective on the temporal stability and diversity of Time-to-Live values associated with domain entries. Furthermore, the analysis extends to zone-related attributes, such as zone entropy, length (Fig. 17), and counts of different DNS entry types, including A, AAAA, TXT, MX, and NS records (Fig. 18). These features collectively contribute to a comprehensive understanding of the DNS infrastructure supporting the domains. Lastly, the gelocation feature investigates the country of origin for each IP (Figs. 20, 19 and 22).

5.1.4 The results

The table (2) presents a comprehensive list of features considered in the data analysis. Each feature is categorized as lexical, IP-based, RDAP-based, DNS-based, or geolocation-based. The table includes the results of the Kolmogorov-Smirnov (KS) test, a statistical method used to assess the dissimilarity between the benign and phishing datasets for each feature.

The KS test compares the cumulative distribution functions (CDFs) of two datasets and yields a statistic (D) representing the maximum vertical distance between these functions. The formula for the KS statistic is given by:

$$D = \max |F_1(x) - F_2(x)|$$

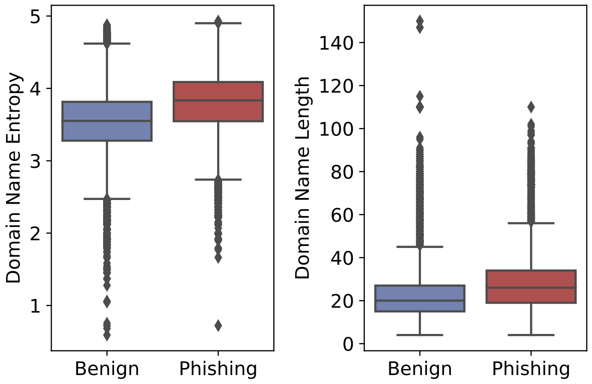

Figure 7. Domain name entropy and length

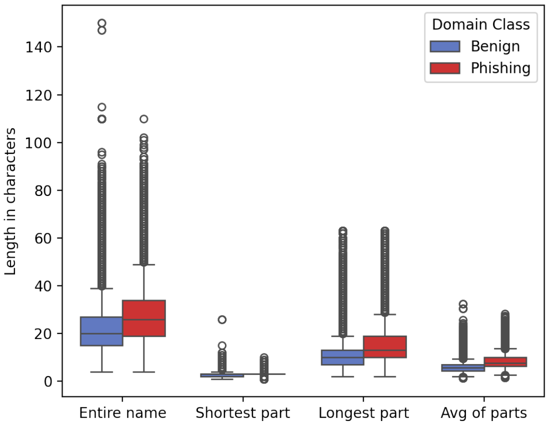

Figure 8. Domain name length distribution

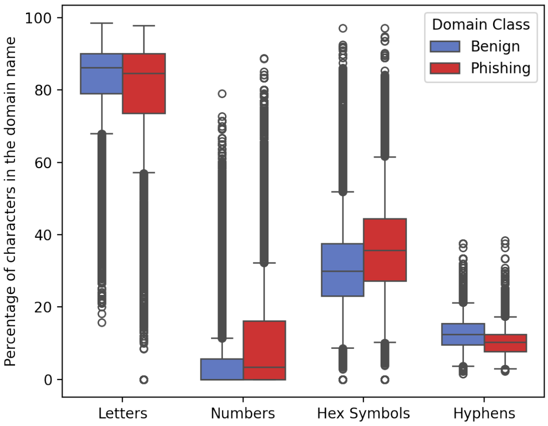

Figure 9. Character distribution in domain names

where x represents all possible values of the variable being analyzed. $F_1(x)$ and $F_2(x)$ are the CDFs of the two datasets. A higher D indicates a larger gap between the benign and phishing domains, while a lower D implies a more similar or overlapping distribution.

To classify the magnitude of difference, thresholds are established:

- **Minimal Difference:** $D < 0.05$
- **Moderate Difference:** $0.05 \leq D < 0.1$
- **Substantial Difference:** $D \geq 0.1$

These thresholds provide a meaningful way to interpret the results of the KS test and categorize the dissimilarity between the feature distributions for benign and phishing domains.

Category	Feature	Magnitude
Lexical	Top-level domain	Substantial
	Domain name length	Substantial
	Domain name entropy	Minimal
	Shortest part length	Minimal
	Longest part length	Substantial
	Average part length	Moderate
	Subdomain count	Minimal
	Char. Letters %	Substantial
	Char. Numbers %	Substantial
	Char. Hex values %	Substantial
	Char. Other %	Moderate
IP based	IPv4 % per domain	Substantial
	IPv4 % of whole dataset	Moderate
RDAP based	Administrative entity	Minimal
	Registrant entity	Substantial
	Registrar entity	Moderate
	Abuse entity	Substantial
	Admin entity	Minimal
	Technical entity	Minimal
	Domain age	Substantial
DNS based	TTL average	Moderate
	TTL stdev	Moderate
	TTL low	Substantial
	TTL mid	Moderate
	TTL distinct count	Minimal
	Zone entropy	Substantial
	Zone length	Moderate
	A record count	Substantial
	AAAA record count	Moderate
	TXT record count	Substantial
	MX record count	Minimal
	NS record count	Moderate
GEO based	Country of origin	Moderate

Table 2. Features Categorized by Origin. Magnitude of difference between between Phishing and Benign datasets included based on KS test.

While many of the features analysed were already included in the final feature vector, some were added based on the results of my analysis.

Some of the newly added features include:

- DNS TLL based featuers
- RDAP Registrar name based features
- ip_v4_ratio

Please note that majority of the figures in this document were subsequently modified by other group members to better meet their vision.

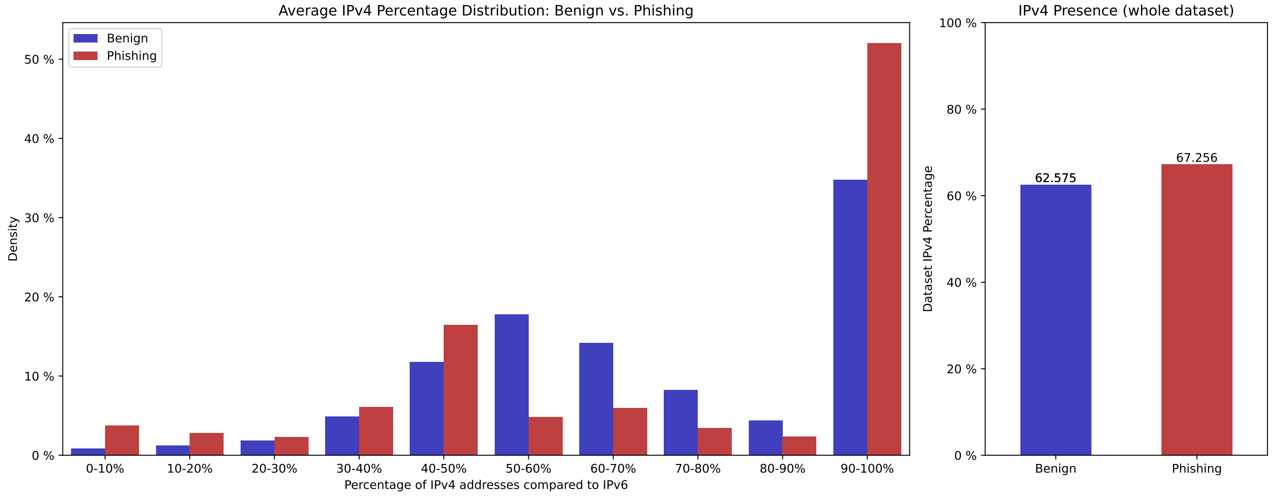

Figure 10. IPv4 presence statistics

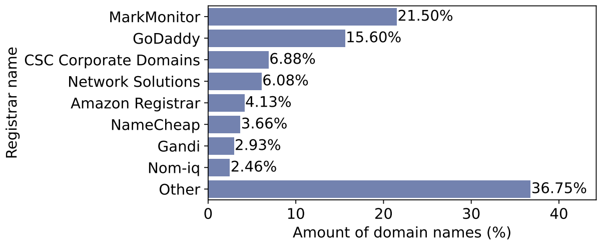

Figure 11. Top 8 registrars of benign domain names (registrar names are shortened)

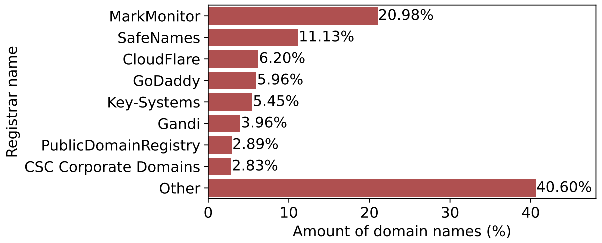

Figure 12. Top 8 registrars of phishing domain names (registrar names are shortened)

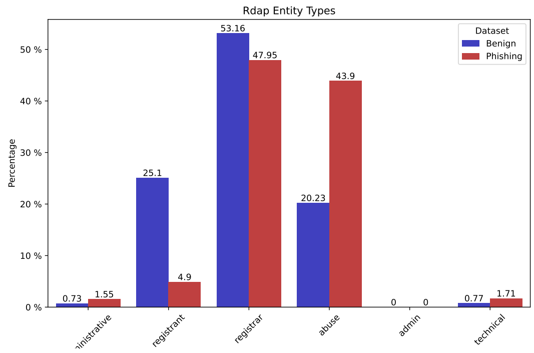

Figure 13. Distribution of RDAP entity types throughout the dataset

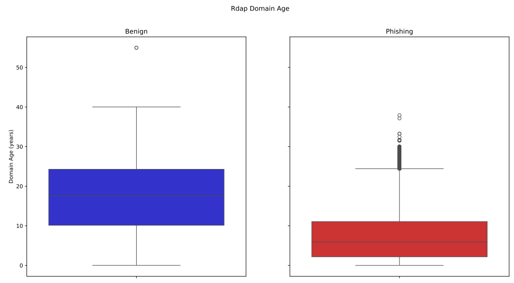

Figure 14. Distribution of domain age in the datasets

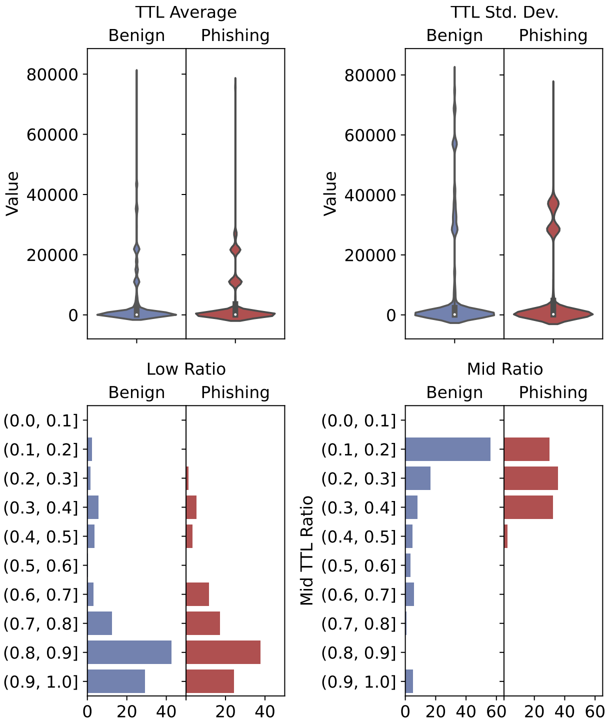

Figure 15. Distribution of domains' vavious TTL properties

5.1.5 Experiments

Besides the conventional way of plotting the distribution of a feature or a set of features, I also experimented with different data analysis methods. Some of them produced satisfying results (as shown by the teaser images on the sides). Although they may not be very useful for a research paper, the resulting plots are quite

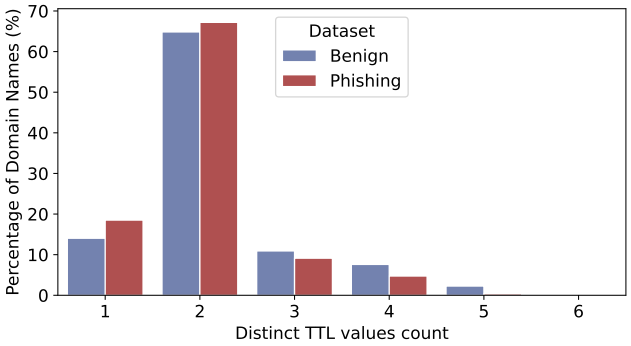

Figure 16. Distribution of domains' TTL distinct values

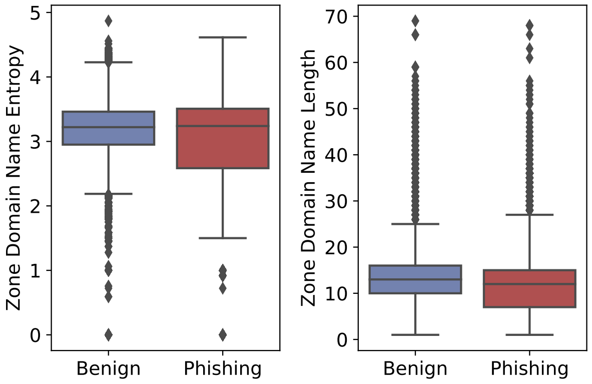

Figure 17. Zone domain name entropy and length

appealing to the eye :).

One of the ideas was to scatter-plot the points generated by two features of both Phishing and Benign datasets on the same "canvas". This creates some plots, where you can clearly see regions that have no overlap, meaning they are characteristic of one of the classes (Phishing/Benign). It's a nice visualisation of a problem, where a regression-based classification algorithm would have an easy time finding these distinct regions.

5.2 Hyperparameter tuning

My second goal was to find optimal hyper-parameter configuration for two decision tree classifiers. The first one was a single Decision Tree, and the second one was its ensemble counterpart, the Random Forest.

5.2.1 Grid Search

To achieve this goal, I employed a systematic approach based on grid search, as depicted in the flowchart (Fig. 21). Initially, wide ranges for all parameters were established to ensure a comprehensive exploration of potential values. Each parameter was tested within its respective range, and the accuracy for each value was meticulously recorded.

This initial exploration served a dual purpose: it provided preliminary insights into optimal parameter values and facilitated the refinement of testing ranges. The latter was achieved by narrowing down the parameter ranges based on the accuracy outcomes of the initial tests. This iterative process of testing and narrowing ensured that each subsequent test was more focused and efficient.

After obtaining reduced parameter ranges, a full grid search was conducted to meticulously explore combinations of parameter values within these refined ranges. This approach ensured that the optimization process was both exhaustive and efficient, balancing the breadth and depth of exploration to identify optimal classifier parameters effectively.

However, this approach was still too slow to achieve any results within a reasonable time frame, due to the large scale of the training data (128 features, ~500k rows). This was probably caused by insufficient resources of my machine for a task of this scale. Therefore, I decided to use a different method, Bayes search, which is more efficient and adaptive.

5.2.2 Bayes Search

Bayes search is an approach that uses Bayes' theorem to direct the search in order to find the minimum or maximum of an objective function [20, 21]. It is an approach that is most useful for objective functions that are complex, noisy, and/or expensive to evaluate [20]. Bayes search uses a probabilistic model to estimate the objective function and a utility function to select the next point to evaluate. By updating the model with each new observation, Bayes search can exploit the information gained from previous evaluations and explore the parameter space more intelligently [20].

5.2.3 The results

The tables below summarizes the results of the hyperparameter tuning for the Decision Tree and the Random Forest classifiers. The best parameters and the test F1 score for each classifier are shown in Tables 3 and 4, respectively. The hyperparameters were further optimized on university servers to achieve even better results, as shown in Table 5).

Parameter	**Value**
max_depth	44
max_leaf_nodes	605
min_samples_leaf	1
min_samples_split	2
Test F1 Score	**0.9326**

Table 3. Hyperparameter tuning results for Decision Tree

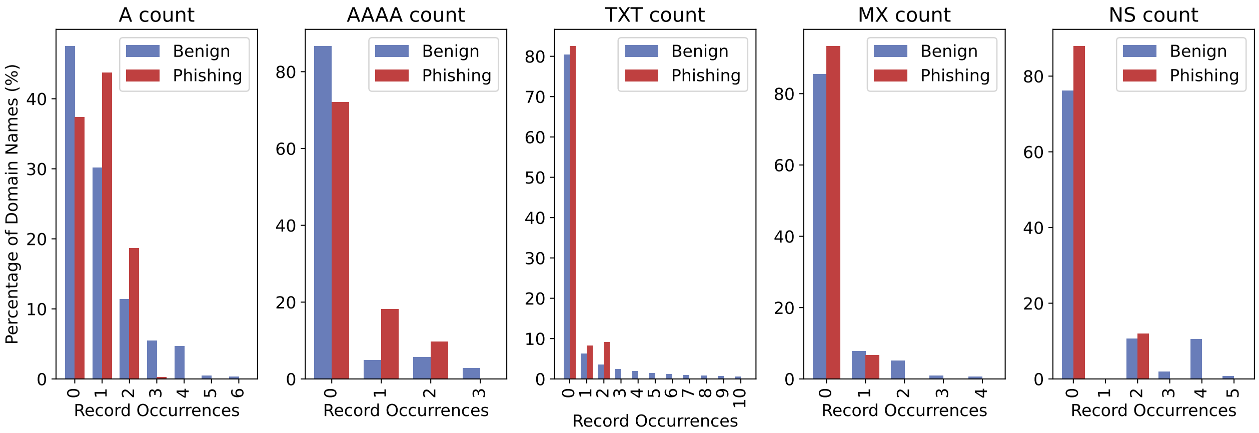

Figure 18. Distributions of record presence of different types

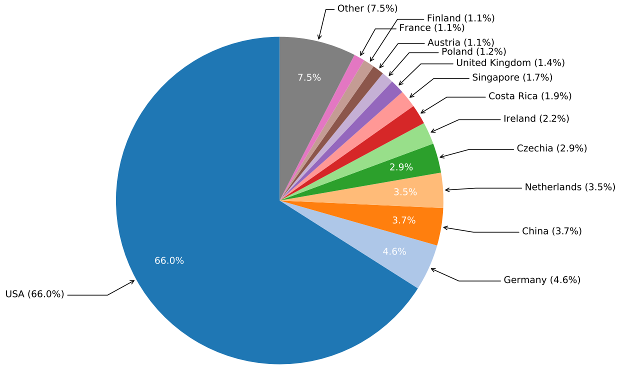

Figure 19. Top countries amongst benign domain IPs

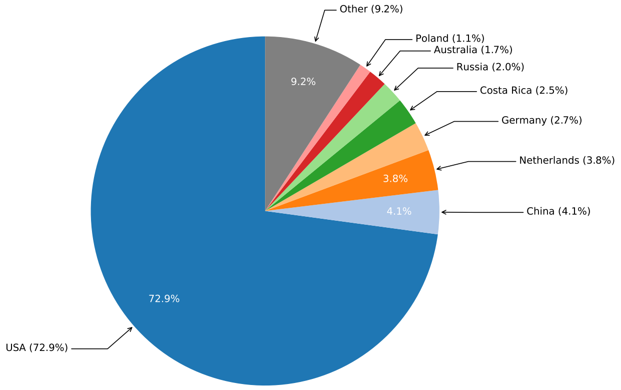

Figure 20. Top countries amongst phishing domain IPs

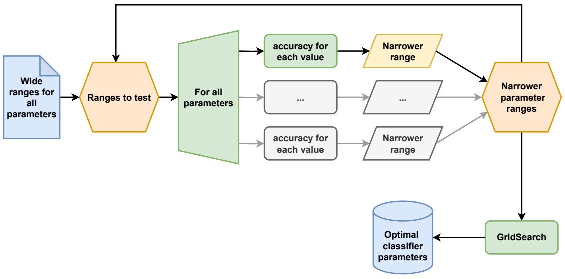

Figure 21. Flowchart describing the process used to fasten the Grid Search

6. Model Execution and Outcome Analysis

Parameter	Value
n_estimators	143
criterion	’entropy’
max_depth	19
min_samples_split	2
min_samples_leaf	7
max_features	None
max_leaf_nodes	174
max_samples	0.5416
Test F1 Score	**0.9352**

Table 4. Hyperparameter tuning results for Random Forest

This section covers the model execution and outcome analysis. We enhanced the feature vector with new features, evaluated various classifiers with different metrics, compared our results with our previous work, and discussed the model stability and consistency.

6.1 Feature Vector Enhancment

To enhance the model’s capacity to discern malicious domains, we incorporated several new features. These additions are based on detailed data analysis and insights gained from the examination of existing models. The following features were added to the original set:

- ’dns_mx_avg_len’
- ’dns_mx_avg_entropy’
- ’dns_txt_avg_entropy’
- ’dns_zone_level’
- ’dns_zone_digit_count’
- ’dns_zone_len’
- ’dns_zone_entropy’
- ’dns_resolved_record_types’
- ’dns_has_dnskey’
- ’dns_dnssec_score’
- ’dns_ttl_avg’
- ’dns_ttl_stdev’
- ’dns_ttl_low’
- ’dns_ttl_mid’
- ’dns_ttl_distinct_count’

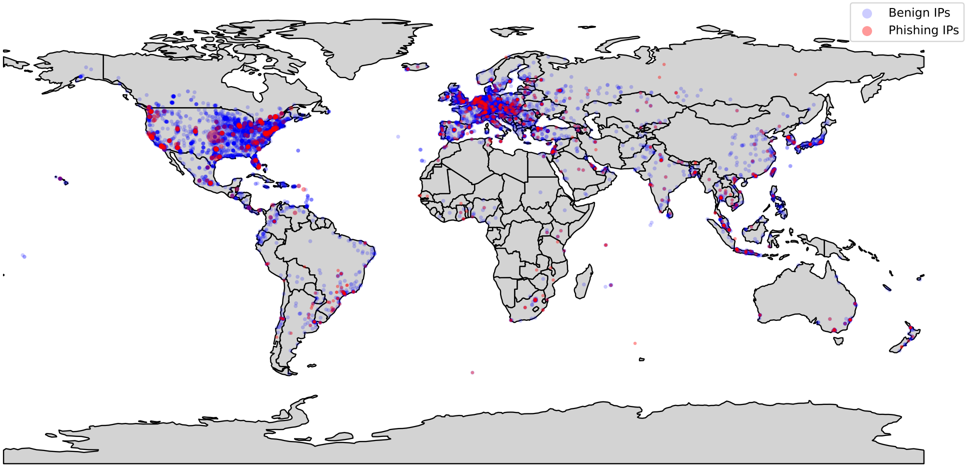

Figure 22. Benign and phishing IP locations

- 'dns_soa_min_ttl'
- 'dns_txt_external_verification_score'
- 'dns_txt_dmarc_exists'
- 'tls_has_tls'
- 'tls_negotiated_version_id'
- 'tls_negotiated_cipher_id'
- 'lex_bigram_matches'
- 'lex_trigram_matches'
- 'lex_sub_vowel_count'
- 'lex_sub_vowel_ratio'
- 'lex_sub_consonant_count'
- 'lex_sub_non_alphanum_count'
- 'lex_sub_hex_count'
- 'lex_avg_part_len'
- 'lex_stdev_part_lens'
- 'lex_longest_part_len'
- 'lex_shortest_sub_len'
- 'rdap_ip_v4_count'
- 'rdap_ip_v6_count'
- 'rdap_registrar_name_len'
- 'rdap_registrar_name_entropy'
- 'rdap_registrar_name_hash'
- 'rdap_registrant_name_len'
- 'rdap_registrant_name_entropy'
- 'rdap_admin_name_len'
- 'rdap_admin_name_entropy'
- 'rdap_admin_email_len'
- 'rdap_admin_email_entropy'
- 'rdap_ip_shortest_v4_prefix_len'
- 'rdap_ip_longest_v4_prefix_len'
- 'rdap_ip_shortest_v6_prefix_len'
- 'rdap_ip_longest_v6_prefix_len'
- 'rdap_ip_avg_admin_name_len'
- 'rdap_ip_avg_admin_name_entropy'
- 'rdap_ip_avg_admin_email_len'
- 'rdap_ip_avg_admin_email_entropy'
- 'geo_lat_stdev'
- 'geo_lon_stdev'

These additions were instrumental in capturing nuanced patterns and improving the model's discriminatory power.

6.2 Evaluation of Results

The evaluation results presented in Table 5 provide a comprehensive overview of the performance of various classifiers. These metrics, including precision, recall, F1 score, and variance, serve as critical indicators of the models' effectiveness in detecting malicious domains.

6.2.1 Comparison with Previous Work

To gauge the effectiveness of our model improvements, we contrast the present results with the performance of the XGBoost classifier employed in our earlier work. The XGBoost classifier was the sole algorithm utilized in the prior study. In that context, the XGBoost model achieved a precision rate of 96.76%, recall of 91.81%, and an F1 score of 94.22%.

The refined XGBoost model in our current implementation surpasses these benchmarks with notable improvements. The F1 score, a critical measure of a model's balance between precision and recall, increased from 94.22% to 97.64%. This enhancement indicates a more robust and accurate identification of malicious domains, demonstrating the efficacy of the feature vector extension and hyperparameter tuning.

6.2.2 Model Stability and Consistency

Apart from improved performance metrics, it is crucial to consider the stability and consistency of the models. The introduced metric of variance provides insights into the degree of fluctuation in the model's performance across different runs. Lower variance values suggest a more stable and reliable classifier.

In our results, all classifiers exhibit low variance values, reinforcing the reliability of our models. Notably, XGBoost and LightGBM stand out not only for their superior performance but also for their consistently low variance. This consistency is indicative of the robustness of the models in handling diverse datasets and further validates the effectiveness of our approach.

7. Conclusion

This report presented the problem of detecting malicious domains on the internet, which poses a serious threat to the security and privacy of users and organizations. The main focus was on two types of malicious domains: phishing domains and DGA domains, which are often used by cybercriminals to launch attacks or control botnets. To address this problem, we proposed innovative machine learning approaches that leverage various features and techniques to improve the detection accuracy and efficiency.

The report demonstrated how we addressed the problem of detecting malicious domains on the internet, from reviewing the existing literature and developing our methodology, to conducting experiments and visualizing the results, to executing and evaluating our models. We also highlighted the development of DomainRadar, an experimental prototype that enhances the threat detection and incident response capabilities. In addition, the report showed my specific contributions in dataset analysis and hyperparameter tuning, which significantly shaped the research efforts and improved the model performance.

The report concluded that our proposed approaches achieved significant improvements in model performance compared to previous work, as well as high stability and consistency. The report aimed to contribute to the field of cybersecurity by advancing the knowledge and methods of detecting malicious domains on the internet.

Moreover, the report reflected my personal learning experience working on a real project that involved machine learning and data analysis. Through this project, I gained valuable skills and knowledge in various aspects of cybersecurity, such as phishing detection, DGA detection, feature engineering, hyperparameter optimization, and model evaluation. I also learned how to collaborate effectively with other researchers and use different tools and techniques to enhance the research process and outcome. I enjoyed working on this challenging and rewarding project, and I look forward to applying what I have learned to future projects.

8. Acknowledgements

I would like to thank my supervisor
Ing. Radek Hranický Ph.D. for his help and guidance throughout this project. I am also grateful to the other research groups members for providing me the references and final versions of the figures used in this report.

References

[1] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2010, pp. 373–382.

[2] S. Gorling, "The Myth of User Education," in *Proceedings of the 16th Virus Bulletin International Conference*, Montreal, Canada, October 2006.

[3] B. A. Gyunka and A. O. Christiana, "Analysis of human factors in cyber security: A case study of anonymous attack on HBGary." *Computing & Information Systems*, vol. 21, no. 2, 2017.

[4] S. Bell and P. Komisarczuk, "An analysis of phishing blacklists: Google safe browsing, openphish, and phishtank," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2020, pp. 1–11.

[5] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," 2009.

[6] C. Lewis and M. Sergeant, "Overview of Best Email DNS-Based List (DNSBL) Operational Practices," RFC 6471 (Informational), Internet Engineering Task Force (IETF), Request for Comments (RFC) 6471, Jan. 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6471.txt

[7] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "Phishnet: predictive blacklisting to detect phishing attacks," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–5.

[8] D. L. Cook, V. K. Gurbani, and M. Daniluk, "Phishwish: a stateless phishing filter using minimal rules," in *Financial Cryptography and Data Security: 12th International Conference, Cozumel, Mexico, January 28-31. Revised Selected Papers 12*. Springer, 2008, pp. 182–186.

[9] E. Kirda and C. Kruegel, "Protecting users against phishing attacks with AntiPhish," in *29th*

	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)	0.987666	5.57e-08	0.965317	1.06e-07	0.976364	5.61e-08	0.001905	1.34e-09
LightGBM (LGBM)	**0.988622**	1.21e-07	**0.966971**	2.88e-07	**0.977677**	1.02e-07	**0.001758**	2.95e-09

Table 5. Comparison of results for individual classification methods explored. The classifiers I contributed to are marked blue.

Annual International Computer Software and Applications Conference, vol. 1, 2005, pp. 517–524 Vol. 2.

[10] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.

[11] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing email detection based on structural properties," in *NYS cyber security conference*, vol. 3. Albany, New York, 2006, pp. 2–8.

[12] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 649–656.

[13] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, 2007, pp. 60–69.

[14] Akamai. (2021) What are domain generation algorithms? [Online]. Available: https://www.akamai.com/glossary/what-are-dgas

[15] 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.

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

[17] 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.

[18] 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.

[19] R. Hranický, J. Polišenský, and A. Horák. (2022) Domainradar: A tool for detecting and analyzing domain generation algorithms. NES@FIT research group, Faculty of Information Technology, Brno University of Technology. [Online]. Available: https://www.fit.vut.cz/research/product-file/742/DomainRadar-documentation.pdf

[20] J. Brownlee. (2020, 9) What is bayesian optimization? [Online]. Available: https://machinelearningmastery.com/what-is-bayesian-optimization/

[21] Wikipedia. (2023, 6) Bayesian search theory. [Online]. Available: https://en.wikipedia.org/wiki/Bayesian_search_theory