UNIVERSITY OF ŽILINA

FACULTY OF MANAGEMENT SCIENCE AND INFORMATICS

DETECTION OF PHISHING WEBPAGES

Study programme:	Applied Informatics
Field of study:	Informatics
Study department:	Department of Informatics
Supervisors:	doc. Ing. Michal Kvet, PhD.

Žilina 2024 **Ing. Ivan Škula**

Abstrakt

Phishing je mimoriadne rozšírená, dynamická, prispôsobivá a nebezpečná forma útoku pomocou sociálneho inžinierstva, ktorý má negatívny cply ako na jednotlivcov tak na celú spoločnosť. Phsihing využíva rôzne elektronické komunikačné kanály. Ako forma útoku je využívaný útočníkmi s rôznou úrovňou expertízy - od príležitostných kybernetických zločincov, po sofistikovaných a technicky zdatných profesionálov. Napriek desaťročiam intenzívneho výskumu sa zatiaľ nepodaril nájsť uspokojivé riešenie, ktoré by phishing vedelo eliminovať. Naša štúdia sa zameriava na nuansy phishingu, zdôrazňuje prispôsobivosť útočníkov, ktorí často prispôsobujú svoje útoky a nasadzujú nové prístupy a metódy. Náš výskum tiež potvrdzuje obmedzenú účinnosť tradičných zoznamov kompromitovaných domén a poukazuje na problémy spojené so získavaním dostatočných a presných údajov. Náš rámec popisujúci process zberu údajov pre prediktívnu analytiku v oblasti phishingu má za cieľ zvýšiť porovnateľnosť rôznych detekčných metód medzi výskumníkmi. Navyše, detailné zachytneý postup návrhu pre system detekcie phishingu v reálnom čase odhaľuje praktické problémy, aj ich riešenia. Náš výskum zachytáva účinnosť najbežnejších algoritmov prediktívnej analytiky a súčasne testuje efektivitu navrhnutých indikátorov detekcie phishingových stránok. Zlepšením techník spracovania údajov si kladieme za cieľ posilniť kvalitu ale aj porovnateľnosť techník detekcie phishingu. Tento prístup nielenže posúva oblasť vpred, ale tiež ponúka praktické usmernenia, ktoré majú za cieľ pomôcť znížiť celosvetový dopad phishingu.

Kľúčové slová: Phishing, Detekcia, Techniky, Ukazovateľ, Obeť, Dáta, Webstránka

Abstract

Phishing is a widespread, dynamic, adaptable, and dangerous form of social engineering attack that negatively impacts individuals and society. It employs a variety of channels and tactics, reflecting the diversity of attackers, from low-skill opportunists to sophisticated cyber criminals. After decades of extensive research, no definitive solution has been found yet. Our study delves into the nuances of phishing, highlighting the adaptability of attackers who frequently deploy new approaches and techniques. This research confirms the limited efficacy of conventional Blacklists and underscores the challenges related to gathering sufficient and accurate data. Our framework for phishing data collection and feature extraction aims to enhance the comparability of different detection methods between researchers. Our step-by-step description of real-time phishing detection solution also uncovers practical challenges and applicable fixes. Our research sheds light on the performance of common algorithms of predictive analytics as well as lists relevant indicators distinguishing phishing webpages from legitimate ones. By improving data processing practices, we aim to bolster the effectiveness and comparability of phishing detection techniques. This approach advances the field and offers practical guidelines that could help reduce the global impact of phishing.

Keywords: Phishing, Detection technique, Victim, Data, Domain

Statement of originality

I certify that I have performed my research as well as the intellectual content of this thesis independently under the guidance of my advisors using my domain knowledge of the subject area along with publicly available information (on the internet or other media) and that all relevant sourced materials are correctly listed in references chapter.

In Žilina, May 1, 2024

Signature: …….

Acknowledgment

I want to express my thanks and gratitude to my advisor, doc. Ing. Michal Kvet, PhD. for his valuable comments and guidance on my research. His insights and unique angle of view on the topics helped me steer my focus toward the most relevant aspects of the study and expand my knowledge in the given area.

Ing. Ivan Škula

Table Of Contents

Abbreviations and acronyms

Abbreviation	Meaning
2FA	2-Factor Authentication
AOL	America Online
API	Application Programming Interface
APWG	Anti-Phishing Working Group
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
AUC	Area Under the Curve
BEC	Business Email Compromise
BPNN	Back-Propagation Neural Network (algorithm)
CDN	Content Delivery Network
CEH	Certified Ethical Hacker
CEO	Chief Executive Officer
ccTLD	Country Code Top-Level Domain
DNN	Deep Neural Network (algorithm)
GB	Giga Byte
FBI	The Federal Bureau of Investigation
FP(R)	False-Positive (Ratio)
HTTP(S)	HyperText Transfer Protocol (Secure)
HTML	HyperText Markup Language
IP	Internet Protocol
IC3	Internet Crime Complaint Center
ICANN	Internet Corporation for Assigned Names and Numbers
ICMP	Internet Control Message Protocol
IVR	Interactive Voice Response
JPG	Joint Photographic Expert Group
KPI	Key Performance Indicator
KNN	K-Nearest Neighbors (algorithm)

MS	Microsoft
MFA	Multi-Factor Authentication
NIST	National Institute of Standards and Technology
OSCP	Offensive Security Certified Professional
OSINT	Open Source Intelligence
OTP	One-Time Password
PDF	Portable Document Format
PNG	Portable Network Graphics
QR	Quick Response (code)
RFE	Recursive Feature Elimination (method)
ROC	Receiver Operating Characteristic (curve)
SLD	Second-Level Domain (URL)
SMS	Short Message Service
SVM	Support Vector Machine (algorithm)
THLD	Third-Level Domain (URL)
TLD	Top-Level Domain (URL)
TP(R)	True-Positive (Ratio)
UAE	United Arab Emirates
URL	Uniform Resource Locator
USA	United States of America
USD	United States (of America) Dollar ($)
VPN	Virtual Private Network
WARC	Web ARChive (file type)
WAT	Web Archive Transformation (file type)
WET	Web archive Extraction of Text (file type)
YoY	Year over Year

Introduction

Phishing is one of the oldest and most common techniques cyber criminals employ. In the U.K., phishing was the most common technique used for 83% of businesses that have identified a breach [24]. In 2022, approximately 4.25 billion email users worldwide communicated through an estimated 333 billion emails daily [35]. If more than half of all emails are spam (53%) [18], and we consider 3% of spam messages as phishing [18], we would get to approximately 5 billion phishing emails sent every day (alternative estimates are a bit lower ≈3.4 billion [70]). With some simplification, we can claim that statistically, every person with an email could be targeted by phishing every day. It is undisputable that phishing is a serious problem that impacts all of us individually and in society.

Though phishing has existed for a long time, its prevalence is steadily and continuously growing. In the last few years - as seen in Table 1 - along with the digital transformation of many organizations and government services [37] - a tremendous increase in phishing attacks has been observed. As a result of these changes, phishing is more common than ever before. It is spread through many new channels like SMS, voice calls, recorded voice messages, chat applications within social networks, and even IVR on top of the more traditional email and web.

Table 1 Number of phishing victims in the U.S. as recorded by IC3 (FBI) [21][22]

	2017	2018	2019	2020	2021	2022	2023
Victims	25,344	26,379	114,702	241,342	323,972	300,497	298,878
YoY %		104%	435%	210%	134%	93%	99%

As depicted in Figure 1, phishing-related research has been increasingly popular among scientists and researchers, with a significant interest spike in recent years. There has also been a gradual increase in research publications focusing on phishing detection techniques [14]. Unfortunately, despite all the efforts and decades of research, no "silver bullet" that would help eliminate phishing or at least significantly reduce its impact has been found. Even after many years of research, it is still the "cat and mouse game" between criminals adapting and perfecting their techniques, trying to achieve their malicious objectives, and technology companies, researchers, security enthusiasts, and government entities trying to prevent and mitigate the impact of their actions.

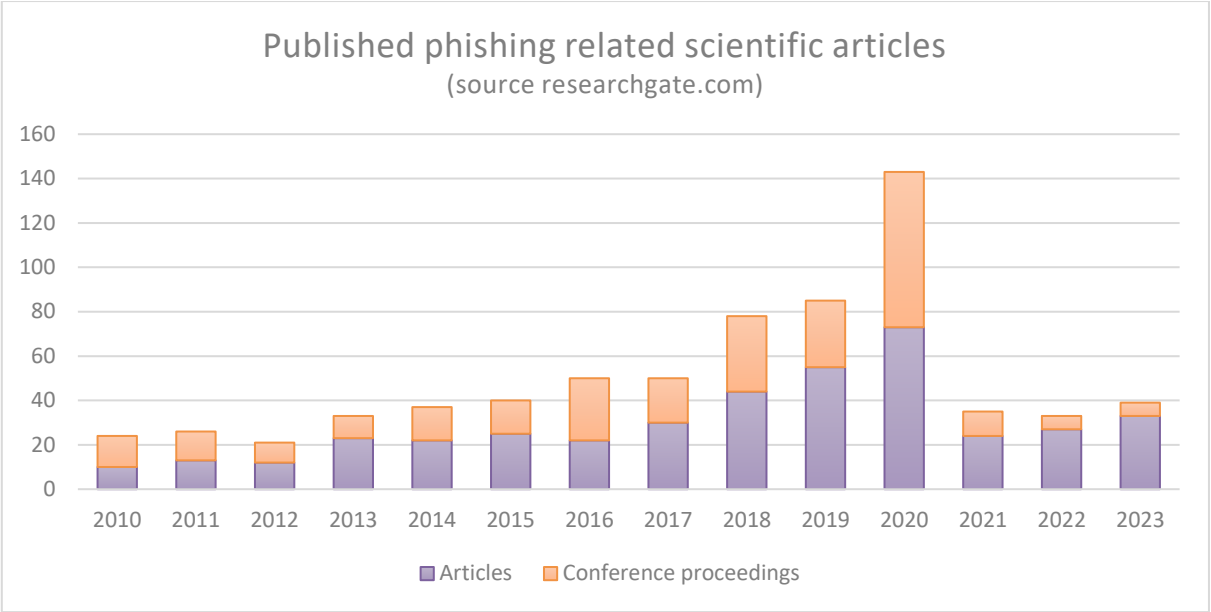


Figure 1 Published phishing-related articles and research papers from 2010 to 2023

It is advised to use a multi-layered approach addressing the users (potential victims) on one side and the technology landscape on the other to reduce phishing and its effects. It often starts with user awareness training, designing and adhering to the security best practices and guidelines, and deploying the technical solutions aiming to detect and prevent phishing. Though the term – phishing – is recognized by the general public (63% of people in the poll answered correctly the question of what phishing is [33]), phishing can have many facets that contribute to the complexity of identifying and preventing it. The complexity of phishing attacks varies and depends on the threat actor's technical maturity, experience, and objectives.

As stated, there is no single and straightforward solution yet. Nevertheless, even minor improvements in existing approaches can meaningfully improve the overall impact of phishing in the current world.

Different technical solutions in the market are trying to tackle phishing through:

- user awareness training supported by simulated phishing messages (this process is intended to continuously monitor the ability of the people to recognize a phishing attack)
- network monitoring solution reviewing and filtering malicious communication

- advanced email filtering solution that flags or directly removes attempted phishing attacks (some autonomously and others with the support of the user or solution operator)
- an endpoint application - browser extensions/plugins [1] or a service agent - that analyzes the webpage characteristics and decides whether the user is trying to access a safe or a phishing webpage

This work focuses on designing and implementing real-time phishing detection solution leveraging predictive analytics algorithms. It was necessary to break down and understand phishing and its building blocks - common use cases, various techniques used to deliver the attacks, and indicators identifying malicious messages or web-pages. The review and analysis of research in this domain allowed us to formulate theoretical and practical objectives and steer the research towards achieving the best-performing detection. Our experiments and research would then address and answer Formulated hypotheses and objectives. The practical part of the thesis describes the step-by-step process of building the solution, starting with data collection, construction of the datasets, training, and comparison of predictive models, and finally, integration of the trained model into the solution – PhishCheck. All these points are comprehensively discussed throughout the work, which is organized as follows:

Chapter 1 initially reviews selected definitions of phishing and their capacity to cover it in full context. Then, it continues with the description of a phishing lifecycle and usual phases, followed by a discussion around the number of stages being used within the attack and differences between the attacks due to the use of various channels, potential objectives, and techniques of the attackers as well as the approach which the attacker chooses - whether targeted or wide-spread attack (so-called spray & pray attack). **Chapter 2** focuses on processes and indicators distinguishing phishing from legitimate web pages or messages. Chapter (2.1) focuses on how humans perceive and identify phishing as it provides a methodology for determining the phishing pages, constituting our research's first theoretical objective. A formalized typology of the phishing page characteristics described in the second part focuses on phishing detection with the help of computer algorithms (Chapter 2.2.4), representing our research's second theoretical objective. **Chapter 3** describes the steps taken to design and implement automated phishing detection solutions leveraging machine learning and

predictive analytics algorithms. The chapter begins with fundamentals around identifying relevant data sources and the data acquisition process. This part – particularly Chapter 3.2.3 – contains a summary of the best practices and solutions to challenges related to phishing data acquisition and data quality adjustments, which fulfills our third theoretical objective. Chapter 3.3.1 summarizes the process and consideration for preparing a comprehensive and balanced dataset that can be used for training the phishing detection predictive model. This summary constitutes the fourth and fundamental theoretical objective of our research. The fifth theoretical objective covered in Chapter 3.4 reviews characteristics and accuracy metrics for different machine learning algorithms and artificial intelligence techniques being used for phishing webpages detection. Though there are claims on the accuracy of the various detection techniques up to almost 99% [1],[26], these can hardly be compared against each other due to the differences in data preparation. Our experiments attempted to establish standard KPI metrics and the pros and cons of each technique when used for phishing detection. The main practical objective of the research - implemented real-time phishing detection solution utilizing a predictive model trained on collected data, which we named PhishCheck - is described in Chapter 3.5. In an attempt to improve the accuracy of the trained model and based on our research around Blacklist's efficacy, we also implemented several versions of Blacklists and Greylists built on collected historical data from various data sources. These are described in Chapter 3.6, and creating these lists constitutes our research's second supplementary practical objective. Our research – achieved results, limitations, and further research suggestions were summarized in the final chapter - Conclusion.

1 About Phishing

One of the most common phishing scams is a “Nigerian Prince email”, “Nigerian letter email,” or "419" (named for violating the given section of the Nigerian Criminal Code).

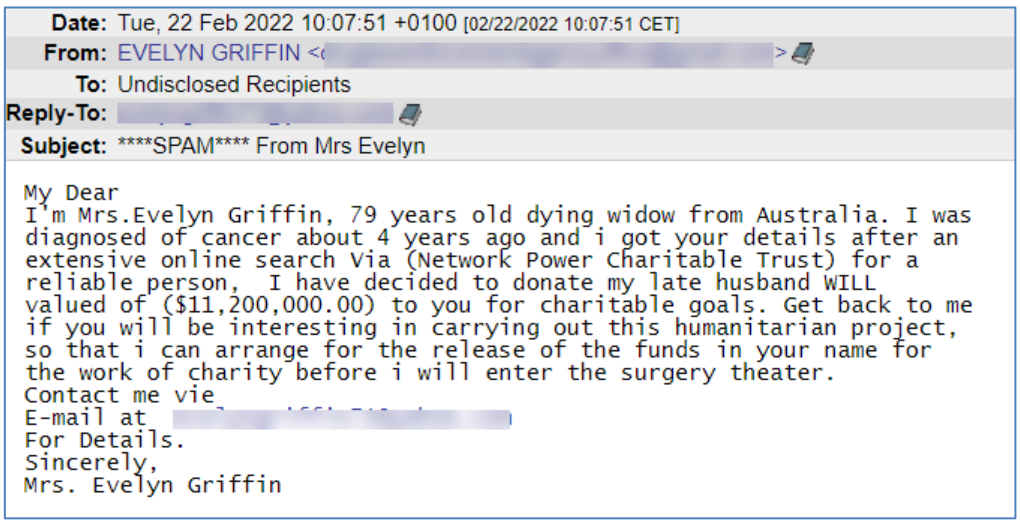


Figure 2 Example email of advanced fee type scam (Nigerian prince email)

In this scheme, the victim receives an email supposedly coming from a Nigerian prince (or royal, businessman, lawyer, high profile person, or another supposedly rich person) seeking help to transfer a large amount of money out of their country (or similar story). Perpetrator most commonly asks for help with money transfers by providing their money for an associated transfer fee, tax, or bribe. In exchange for their participation, the victim is promised considerable money. This scam is hundreds of years old and, in the fraud typology, is also known to belong to the group of scams called an advanced fee scam. Based on this example, we could define phishing as:

"Phishing is a deceptive email sent to a victim to gain their trust and obtain a financial benefit. "

Though this statement describes the oldest and most common form of "phishing," and it would be accurate in the early days of phishing, in today's reality, it doesn't fully capture the variability of phishing in its full context.

The Cambridge Dictionary [49] describes phishing as

"phishing is an attempt to trick someone into giving information over the internet or by email that would allow someone else to take money from them, for example by taking money out of their bank account"

This definition does not consider an option where the phishing is not performed over the internet, e.g., voice phishing (also known as vishing). And even though stealing the victim's money is the most common objective, it is not always the case.

Wikipedia [50] describes phishing as

"phishing is a form of social engineering and scam where attackers deceive people into revealing sensitive information or installing malware such as ransomware"

Perpetrators can indeed send an email, but today, they can also easily use other electronic channels. The attacker can use SMS, voice messages, QR codes, or voice calls to conduct a phishing attack. Also, when we look at the objectives of the perpetrator – it doesn't necessarily have to be only money they are after; it could also be gathering credentials to access different systems or credit card details. Phishing can also be a first step in gaining access to the company's internal network through collected credentials via file-less malware or zero-day exploits. Phishing is indeed a multi-faceted problem, and we will break it down in Chapter 1.2 in more detail.

To describe phishing in its broader context, we say:

"Phishing is a deceptive communication, conducted through electronic channels, from the perpetrator towards the victim where the perpetrator is trying to establish trust and gain benefits which victim wouldn't give up or provide unless deceived or extorted. "

1.1 Short History Of Phishing

Phishing first appeared in the mid-90s when its first occurrence took place along with a spread of the internet in the U.S. (e.g., provided by AOL). Almost 30 years later, phishing is still there and has become one of the most common cybercrime techniques [21]. The first

occurrences of phishing (around 1995) used email to imitate messages from administrators or users of authority. The objective was to deceive the victim into providing user access credentials (this was the case with the early AOL phishing scam [34], where the collected credentials were used to access the internet).

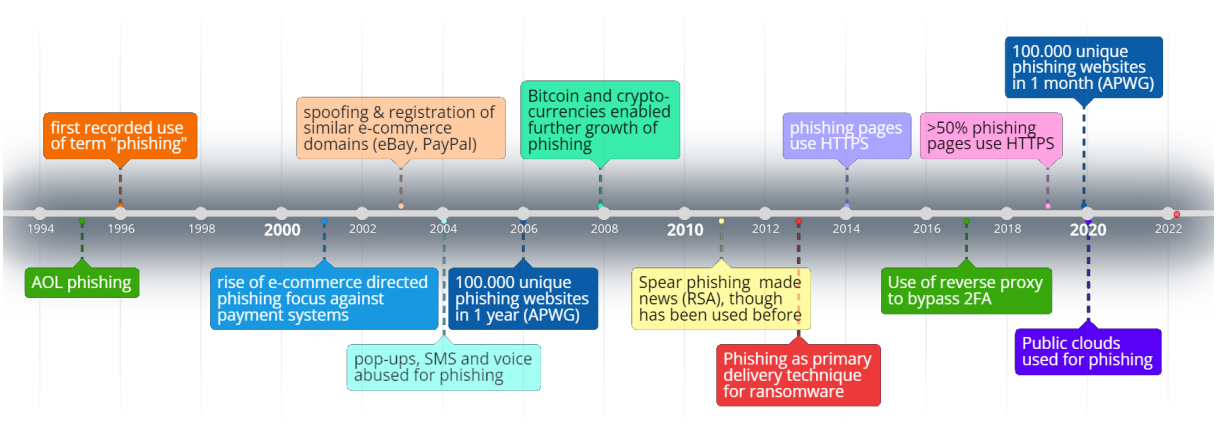


Figure 3 Overview of selected main historical events related to phishing

At the same time, two other famous companies were founded – Amazon, which started as an online books store, and AuctionWeb, which would become the most prominent auction web in the world, known as eBay. A few years later, in 1998, Confinity and x.com were founded, only to become well-known PayPal after their merger in 2000. These are the most notable examples of e-commerce behemoths who helped spark e-commerce to new and unprecedented heights. The growth of e-commerce attracted new types of phishing, which would imitate famous sites (eBay, PayPal, and AOL were among the top 5 most spoofed webpages between November 2003 and January 2004 [4]). At around the same time, in November 2003 in San Francisco, the first meeting of the newly established Anti-Phishing Working Group took place. APWG is an international coalition that brings together different organizations (private or public), government, and law enforcement agencies to reduce and mitigate the impact of fraud – primarily focused on phishing.

Around 2004, new electronic channels became attractive for conducting Phishing – SMS and voice. Both are related to the widespread use of cell phones during the new millennium's first decade. As a result, two new terms related to phishing were born – smishing and vishing. Smishing is phishing conducted through SMS; vishing describes phishing performed through voice calls. In 2006, APWG reported for the first time that 100000 unique phishing websites

were recorded within one year [5]. This same volume milestone would be achieved and more than doubled in August 2020, as the figure captured per month [7]. The actual number would be 201591 unique phishing websites.

Another significant shift happened after 2008 with the arrival of crypto-currencies, which were the catalyst for the overall cybercrime landscape in the years to come, allowing instant and (supposedly, due to the publicly available ledger and public availability of all historical transactions) hard-to-trace financial transactions between the victim and the attacker.

Since the early days of phishing, the most common use case has been direct financial gains, while other objectives have been rare and mainly used by state-organized cybercriminals. In 2011, a spear-phishing attack against RSA Security (a computer and network security company) exposed the danger of focused phishing against particular users. In the attack, 4 RSA employees (unrelated to I.T. or holding high-value access privileges) received a targeted phishing email containing a malicious attachment. This file (MS Excel spreadsheet) used a zero-day exploit in Adobe Flash, allowing attackers to gain access and privileges [28]. As part of the spear-phishing attack, the message is customized and fine-tuned specifically for the recipient. The message must appear genuine, and that's why personal details are often used. Conducting a spear-phishing attack became much easier after the global adoption of Facebook and other social networks since 2010 [41]. These websites are a proverbial gold mine for the attackers as they can profile the victim and customize the spear-phishing message with precise details made public by the victims themselves.

A few years later, in 2013, phishing became the most common delivery method for Ransomware [45], thanks to the overly successful Cryptolocker attack. Around 2014, phishing pages started to use HTTPS protocol instead of standard HTTP [10]. This was mainly aimed at making the user accessing the site appear safe and legitimate. Using secured HTTP (HTTPS) protocol instead of HTTP only ensures that the communication between the user's computer and the web server where the website is hosted is encrypted and, even if intercepted, the ongoing communication can't be easily understood as opposed to cleartext transfer via HTTP. This, unfortunately, has nothing to do with the malicious intent and actual purpose of the phishing website itself. HTTPS only relates to the underlying technical aspect of network

communication. In the first quarter of 2019, over half of the phishing websites used HTTPS [6],[19].

With the increased usage of MFA (**Figure 4**), attackers face a problem. It is insufficient to collect only the user's login details to access a computer or online services. When MFA is activated, the user usually has to provide additional security details in addition to the credentials. This further detail might be provided as an SMS message on a registered mobile number, an email sent to a registered email address, a token available on the mobile application, or even a voice call through IVR. This additional measure becomes a problem for the attacker as he/she might not have access to any of the above.

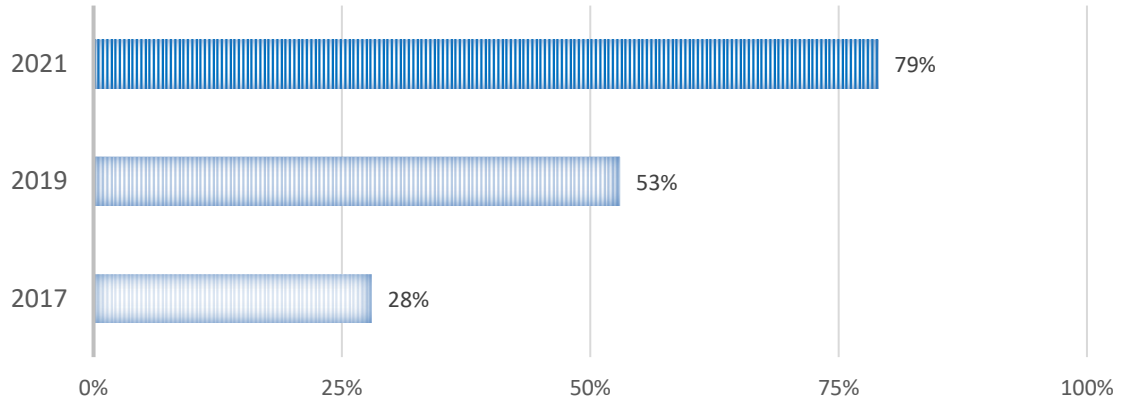


Figure 4 How many users in the poll have used 2FA? [17]

Though MFA significantly enhances the overall security posture of the user, there are solutions to bypass it. One solution is a transparent reverse proxy (**Figure 5**). The attacker creates this proxy and hosts it on a server he controls. The single purpose of this proxy layer is to listen to the communication between the victim and the genuine website after the victim is directed to the proxy server URL from the phishing message. The proxy server captures the credentials provided by the victim and a session cookie returned by the genuine website after the victim provides all verification details. An attacker can later use these details to connect to the legitimate website using the victim's cookie, even without access to 2FA or MFA or knowledge of the password.

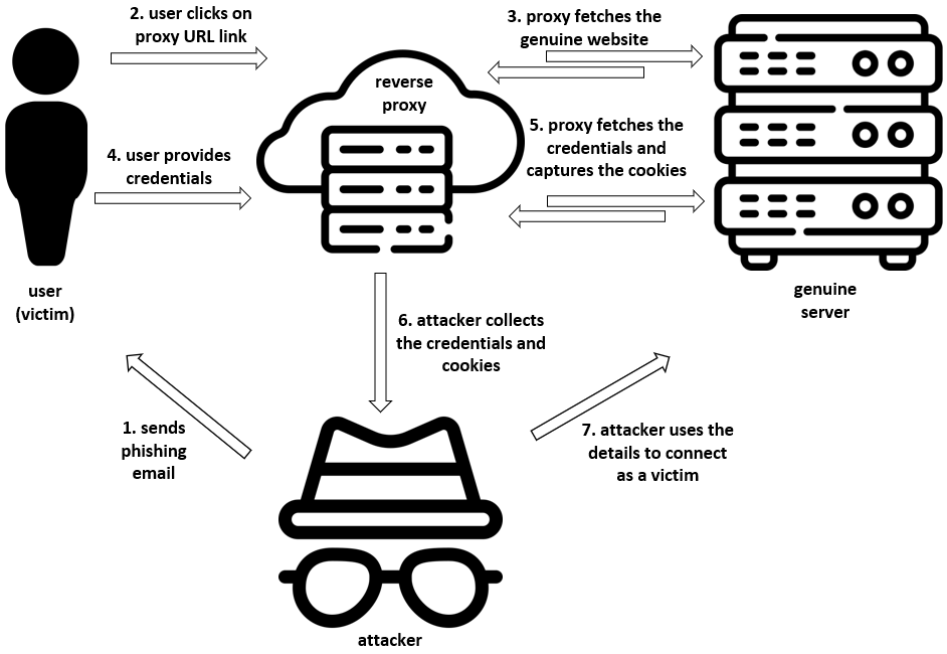


Figure 5 Steps of the phishing attack when using a reverse proxy to bypass 2FA

As the techniques to detect phishing websites and domains are getting more comprehensive, it is harder for the attacker to pass through the layers of protection and expose the victim to the phishing landing page.

Different solutions look into the domain details of the URL and can quickly identify newly registered domains that pose a higher risk of phishing. Similarly, a phishing Blacklist can identify specific IP addresses or domains which might have been linked to some previous phishing attacks. When attackers want to bypass the phishing detection algorithm, they can attempt to host their phishing page on a trusted domain. They could do this by taking over the domain or subdomain. However, it requires a particular technical knowledge to exploit the existing known vulnerabilities of the web server hosting the web page or try to find a new one within the web application itself. Another way to make phishing webpages appear safe is to host malicious pages or code in the public cloud architecture. Public cloud IP addresses and domains are auto-assigned and usually considered safe from phishing detection algorithms. One of the more recent trends is to utilize public cloud infrastructure, storage, or services to hide the malicious nature of phishing websites.

Phishing is very adaptable and has proven to be very dynamic. While the old phishing techniques used in the early days are still being used, many new variants are introduced

immediately as the latest technology is proven and provides some favorable characteristics that perpetrators could leverage.

1.2 Anatomy Of The Phishing

Phishing can be very elastic, and it is impossible to fit it into predefined boxes in a truly comprehensive manner. Let's assume the perpetrator decided to harvest the credentials. In such a scenario, a phishing email might only be the first step of the attack, followed by the spoofed webpage, which will collect the victim's login details. On the other hand, if the perpetrator is attempting to gain access to the victim's computer or user's account, an email with a malicious attachment might be all that is needed.

Many phishing characteristics are closely related; for example, when an attacker chooses a threatening approach, it immediately narrows down the context in which such a method would be acceptable – e.g., police or tax authority, etc. Similarly, when an attacker wants to focus on a particular population – e.g., senior people, he might consider a more favorable channel, e.g., a voice call (voice phishing would be more efficient than an email in this scenario).

1.2.1 Phishing Attack Lifecycle

Phishing attacks, as depicted in Figure 6, usually start with an initial phishing message passed through one of the electronic channels (stage 1): email, SMS, voice calls, social media, instant messaging applications, or QR codes. The list of channels used for phishing is continuously growing as the general public is actively using more electronic channels. This first step is usually needed to direct the victim to the pre-built phishing landing page (stage 2) (through the provided link in the message), which is often an imitation of a genuine website. Sometimes, a phishing email doesn't have any link to the landing page but might contain a malicious attachment (though only 24% of phishing emails include an attachment [40]) that, if accessed, will fulfill the attacker's necessary objective.

Depending on the objective of the phishing attack, the victim, even just by accessing the phishing landing page (stage 2), might be in danger as it might already contain malicious code (stage 2, bottom pictogram); for other objectives, the victim might be asked for their

credentials (stage 2, top pictogram), or payment card details or even for payment through crypto-currency or a gift card after opening the landing page (stage 2, middle pictogram). This is usually the last step of the phishing attack, during which the victim might still repel the attack. If, during this stage, the victim follows through and submits the payment card details or credentials, the phishing attack is successful. Stage 3 depicts the post-attack steps, where the obtained information is used to collect the "reward." As a result, the victim might a) suffer a financial loss (directly or as a consequence of the credentials theft), b) in some cases, the attacker might steal data, sensitive information, or intellectual property, which separately or in combination might result in c) reputational loss.

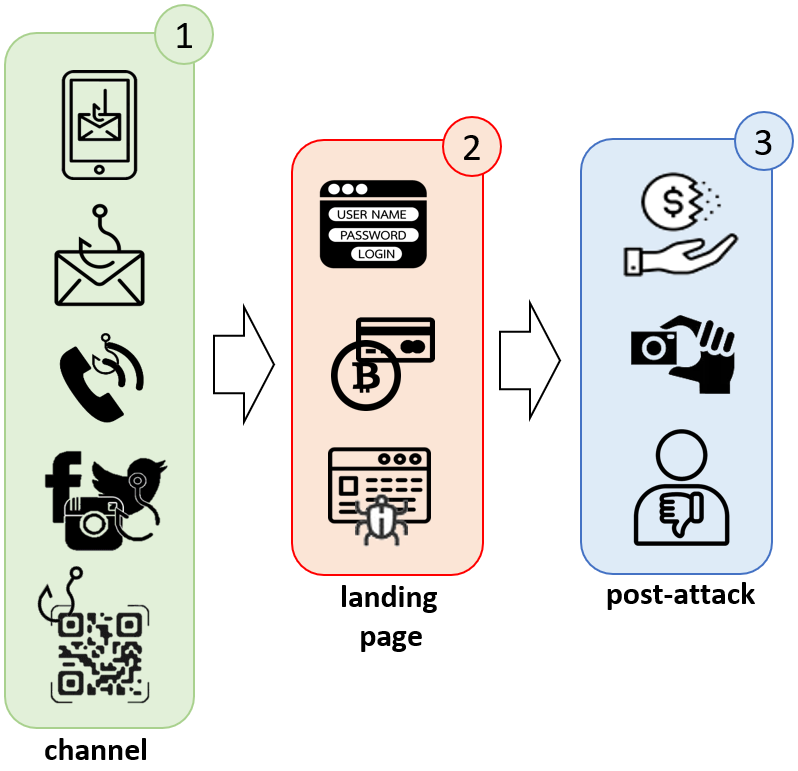


Figure 6 The most common stages of the phishing attack

However, not every phishing attack has to follow the above-prescribed flow from the start to the end. Some phishing attacks might start directly from the phishing landing page (stage 2) and entirely skip the initial phishing message (stage 1). For example, a pop-up window with a phishing landing page might open while browsing the web. Other attacks might direct the victim to the landing page (stage 2), which might be just hosting a malicious file that will allow the attacker to reach the objectives, and the victim doesn't even have to provide any details. Finally, in the case of vishing, an attacker might not even use the landing page

(stage 2) but might try to convince the user (e.g., using the pretext of being technical support) to let the attacker access the victim's computer remotely.

1.2.2 Categories Of Phishing

Phishing attacks differ by the channel used to conduct the attack and the perpetrator's objective. They apply different social engineering techniques and vary by the level of focus on the victim. An important aspect is the number of stages (expected workflow through which the victim would go). The main categories and subcategories are listed in **Figure 7**.

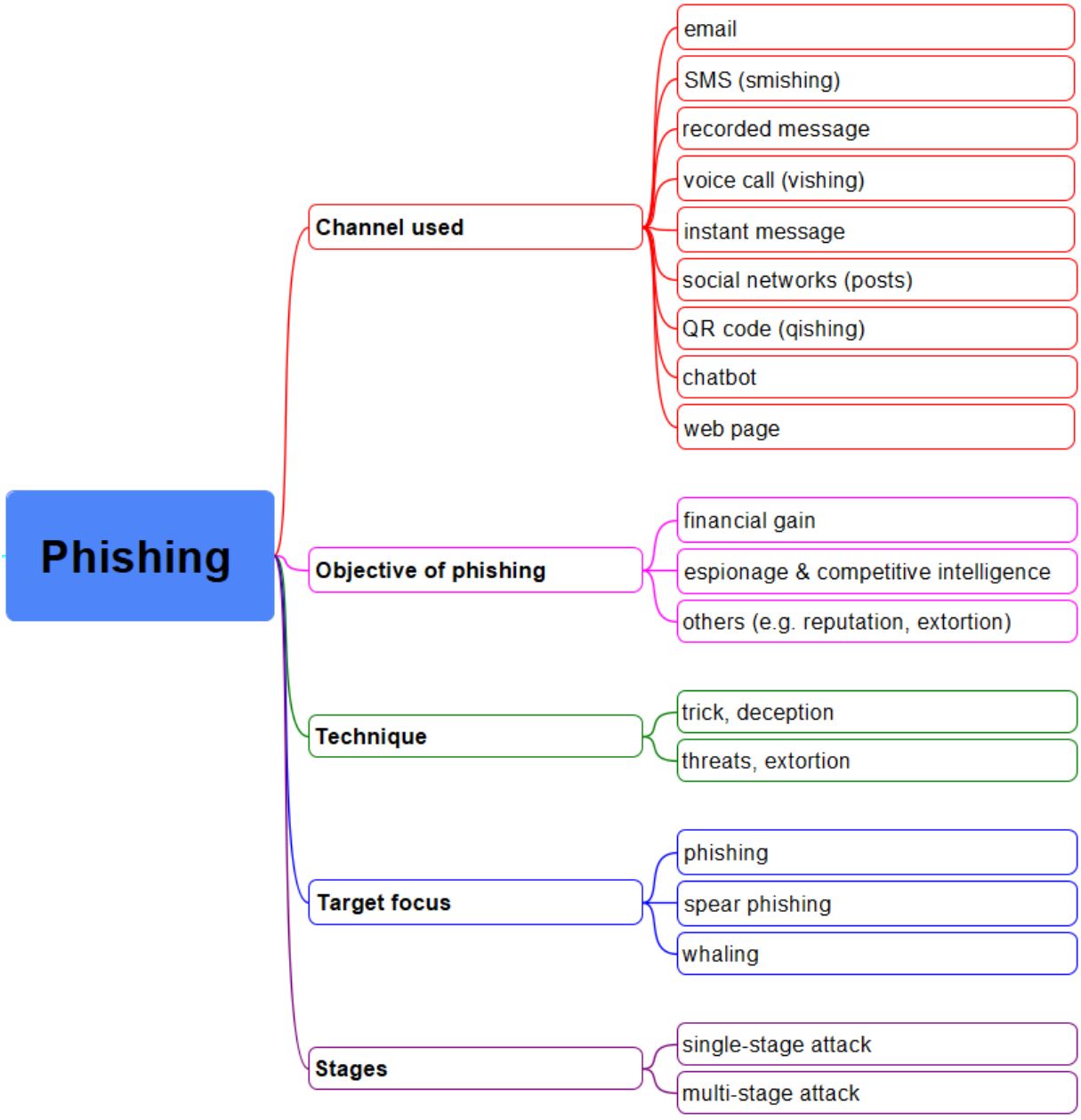


Figure 7 Phishing categorizations by selected characteristics

These are not all the angles we can use when analyzing phishing. Different angles could be, e.g., the geographical focus of the phishing (focused on a specific geography or country and specific language), the intended duration of the phishing attack (short and linked to a particular season or holiday, or universal phishing applicable at any time of the year), or the type of imitated company (e.g., attacker imitating postal service office or global delivery companies). DHL was the most commonly imitated brand for phishing, with 23% of all phishing attacks globally in Q4/2021 [12].

1.2.2.1 Categories By Channel

The first phishing attacks used email as a communication channel, and it has been doing so till today. Email is the most common channel used for phishing attacks, but other electronic channels are also utilized for phishing campaigns (see **Table 2**), and their share is growing. 75% of organizations experienced at least one attempt of non-targeted email phishing attacks (this number does not include spear-phishing or business email compromise scenarios). SMS phishing (smishing) was spotted by 60% of organizations, and voice phishing (vishing) was identified by 53% of organizations [33].

Table 2 Number of recorded phishing attempts by category and channel [33]

		Number of recorded attempts in 2021							
Category	Channel	Unknown	None	1-10	11-25	26-50	51-100	100+	> 1
phishing	**email**	2%	23%	34%	17%	11%	7%	6%	75%
spear-phishing	**email**	1%	34%	29%	15%	11%	7%	3%	65%
BEC	**email**	1%	35%	27%	16%	10%	8%	3%	64%
vishing	**voice**	1%	46%	21%	12%	10%	6%	4%	53%
smishing	**SMS**	1%	39%	24%	12%	13%	7%	4%	60%
social network	**web**	2%	39%	25%	13%	11%	6%	4%	59%

Email is the most used channel by far, whether we look at non-targeted phishing attacks or even if we look at spear phishing, or business email compromise (BEC) attacks. Smishing, though, could be considered even simpler (limited length of the message and uses only text) but bears some cost, as sending SMS is not for free. Due to the fixed size of SMS messages, users can't always see the full URL link or shortened links are used, which increases the susceptibility to this form of phishing [31]. Vishing requires verbal social engineering skills to

be successful, which reduces the pool of perpetrators and eliminates the majority of opportunistic fraudsters. That is why the number is the lowest compared to all the other channels presented.

QR codes as a channel are the most recent addition to the existing mix of channels abused by cybercriminals. Phishing conducted via QR code has been named qishing. QR codes are a “perfect vehicle” for phishing attacks as they are practically impossible for a person to understand and assess the message encoded into it, particularly in phishing – what URL is being represented by it. The potential victim who is shown a QR code wouldn’t know the destination URL and would be inclined to scan it (especially the visually attractive QR codes generated with the help of generative AI as depicted in Figure 8, or the ones placed at places where QR codes are commonly used - e.g., restaurants or information kiosks, etc.).

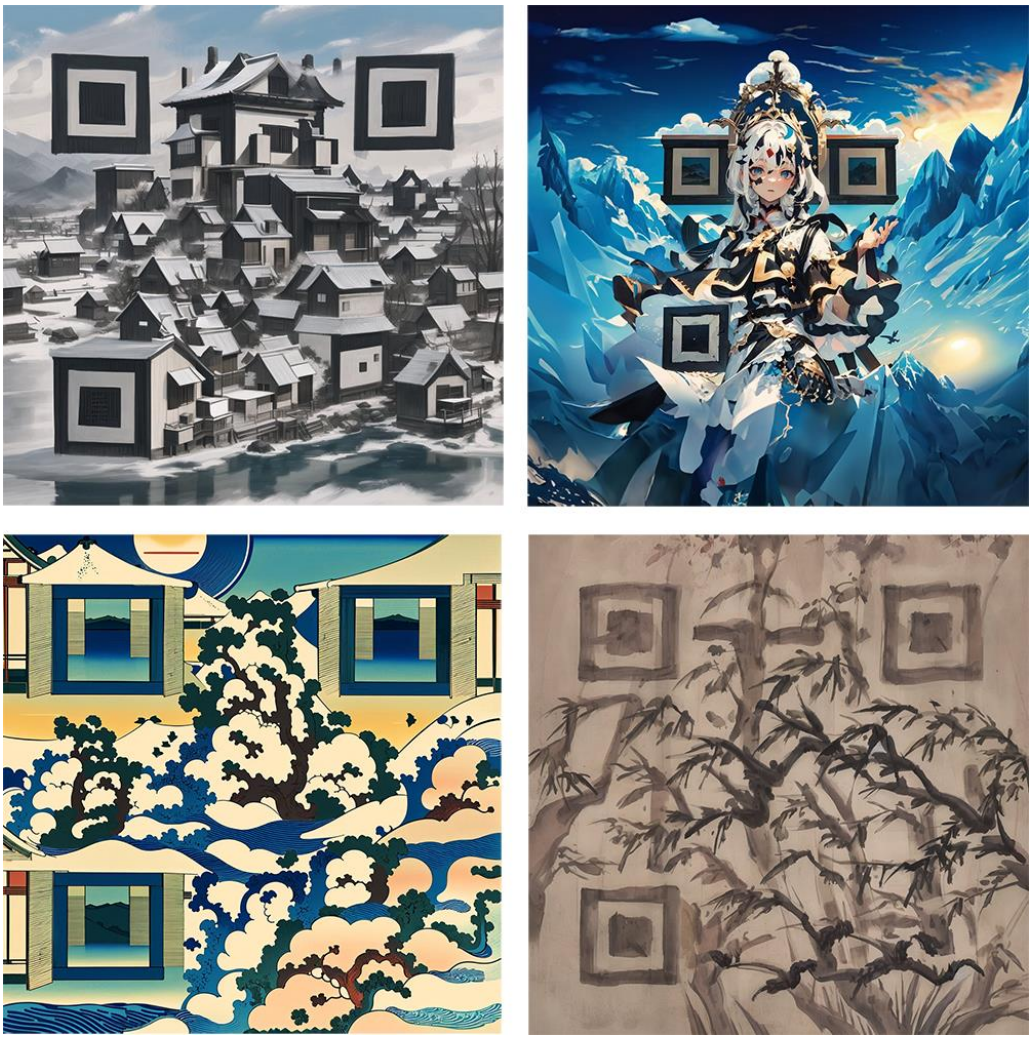


Figure 8 QR codes created with the help of generative AI

Channels and the losses incurred:

From a losses perspective, the highest-ranking category [21] – Business email compromise (BEC) – only in the U.S. has caused more than 2.3 billion USD losses. BEC is a scheme where the attacker uses a spear-phishing attack against staff (usually) in the accounting department. The attacker is impersonating a managing director, CEO, or manager from the highest ranks within the organization's hierarchy and, usually via email, tries to push for payment of an invoice into a specific account, utilizing all the usual techniques (position of superiority, time criticality, even a form of pressure) to ensure the payment is processed at earliest. Just this single scheme makes email the most impactful channel of all.

Vishing is best represented through the "Tech support" scheme, which accounted only in the U.S. for losses above 300 million USD in 2021 [21]. This scheme is pervasive in Southeast Asia and, most commonly, in India. Due to their English proficiency, attackers call selected or random phone numbers in the USA, and not rarely are they trying to target particularly older people. They pretend to be calling from a Microsoft or other commonly known company. The pretext being used is that of technical support staff, who became aware of a virus or a technical issue on the victim's computer and are calling to try to help secure the computer or remove the infection. These are just some of the variants of this scam; there are many more, but the objective is the same. To gain the victim's trust and make them install a remote access application to access the victim's computer directly. Then, they try to get the victim to buy gift vouchers of high value or ask for access to Internet banking so they can deposit money, while in reality, they will be stealing the money from them. It is a simple scheme preying on people with limited knowledge or experience.

Multiple schemes could be performed through social networks – the most obvious is the trust/romance scam, but cryptocurrency-related scams are also widespread. According to [22], almost 2.5 billion USD was lost in 2022 to cryptocurrency-related scams. The losses incurred through social networks were more than 200 million USD in 2021 [21] and 235 million USD in 2022 [22].

Factors contributing to the usage of a given channel for phishing:

- **Channel usage among the general public** – the more familiar the general public is with the channel, the more established it is throughout the different social and age groups, and the easier it is to reach the relevant victims through it.
- **Anonymity on the channel** – higher anonymity of a particular channel increases the chances of its adoption for phishing. Email is a perfect example, as it's almost impossible to link an email address to a specific person. In addition, emails (especially private ones) are often protected only with a password, making them susceptible to potential hacking and abuse.
- **Low or zero cost for channel usage** – the lower the channel's cost, the higher the chances it will be used for mass-distributed phishing campaigns. Email or instant messages are examples of media where the price is practically zero. Also, creating another email account or instant messaging profile is free.
- **Easy automation and re-usability** – the easier it is to automate the channel communication, the higher the chances the channel will be used for phishing. This is especially true for mass-distributed phishing campaigns. On the other hand, these channels usually bear lower efficiency.
    - every step of phishing can be automated and, as such, requires minimal human efforts
    - there are already open-source tools automating the whole process

The most commonly used channels for phishing align very tightly with the above characteristics, making them very efficient tools for conducting phishing attacks.

Each channel has its pros and cons. Though some channels might require more skills on the perpetrator's side (e.g., vishing, where the perpetrator has to communicate and use his social engineering skills) than others, among the benefits could be access to more vulnerable victims (elderly or less privacy-aware people) and, therefore, higher chances of success.

1.2.2.2 Categories By Objective

There are different objectives that the attacker might be trying to achieve. The most obvious one is financial gain. Most attacks (86%) are financially motivated [44]. This group

contains attacks where victims' credentials are used directly to steal money from their financial institution account (with or without their awareness). Another common type of attack is vishing, where the victim is convinced to send the money via bank transfer or by purchasing a gift card. Extortion phishing attacks are another example where a victim practically pays a ransom due to successful extortion. The last but rapidly growing type is via cryptocurrency wallet – where the victim loses money held in a cryptocurrency. This type of attack has been growing mainly in recent years, where in [56], based on data from May 2021 to April 2022, an increase of 257% in targeting cryptocurrency brands was recorded compared to the previous twelve-month period.

Approximately 10% of the breaches are assigned to the espionage and competitive intelligence category [44]. An example of such an attack is called "Operation Shady RAT," when a spear phishing email with an exploit was sent to selected individuals with the desired access privileges within the organization. This would allow attackers access to the internal network. In this attack, 32 organizations were successfully breached, including government organizations from various countries, companies from different industries and defense contractors, non-profit organizations, and a think tank [72]. Extremely alarming is this statement from the whitepaper: "V*irtually everyone is falling prey to these intrusions, regardless of whether they are the United Nations, a multinational Fortune 100 company, a small, non-profit think tank, a national Olympic team, or even an unfortunate computer security firm.*" [72]. This statement only underscores the immediate need to improve existing phishing detection techniques, which are of the highest importance.

The remaining objectives combined represent 4% of the attacks [44]. There might be intermediary objectives within these categories, like obtaining credentials, MFA codes, access to a computer, collecting private or sensitive data from the victim, etc. Those are rarely the ultimate objectives, and though they could be monetized separately (e.g., selling the credentials on the darknet), they usually serve only as a means to reach the final objective.

1.2.2.3 Categories By Technique Applied

When preparing for a phishing attack, attackers have two options when considering the technique to achieve their objective. The first option is to trick or deceive the victim, and the second is to threaten or extort the victim.

To achieve the objective through trick or deception, the attacker has to define a suitable pretext for the attack – for example, an email imitating a familiar parcel delivery company, DHL, DPD, or others and asking for a delivery service fee payment so the pending parcel can be delivered. The attacker will have to focus on all aspects related to the message - the source domain from which the email will be sent and the email formatting to look genuine (using the logo, simple technical text, maybe even imitating a corporate webpage). Another step would be to prepare the phishing landing webpage, to which the email link will navigate the victim. It must appear as a legitimate DHL webpage with a form for filling in the card payment details. For the attacker, it is essential to focus on mitigating all possible aspects that might alert the user and raise suspicion.

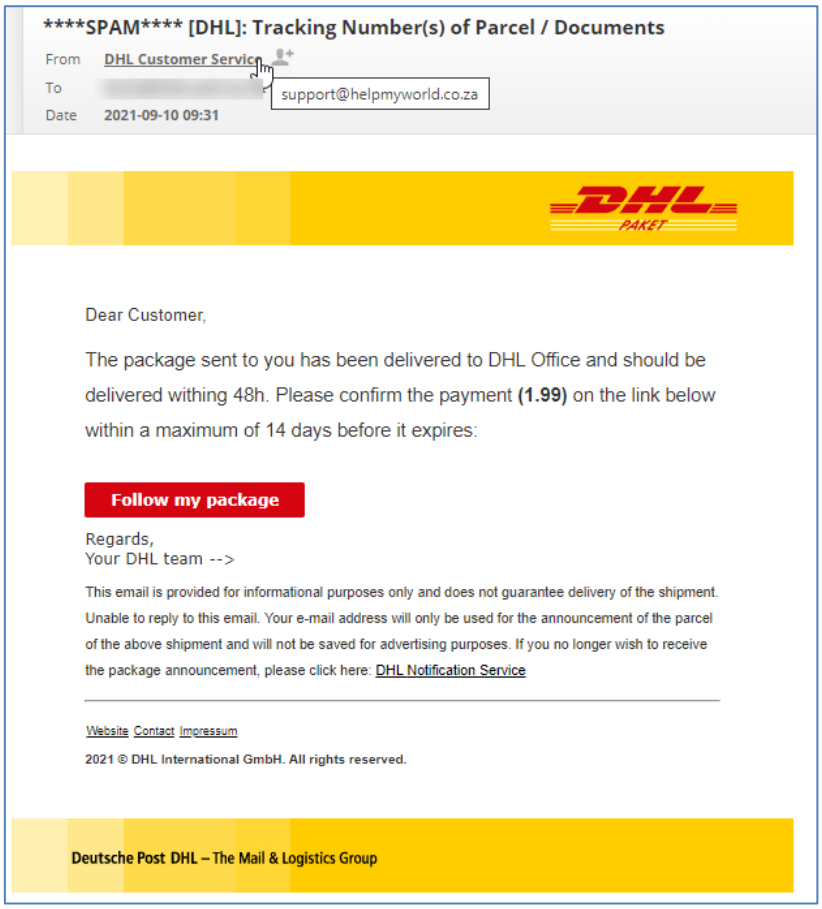
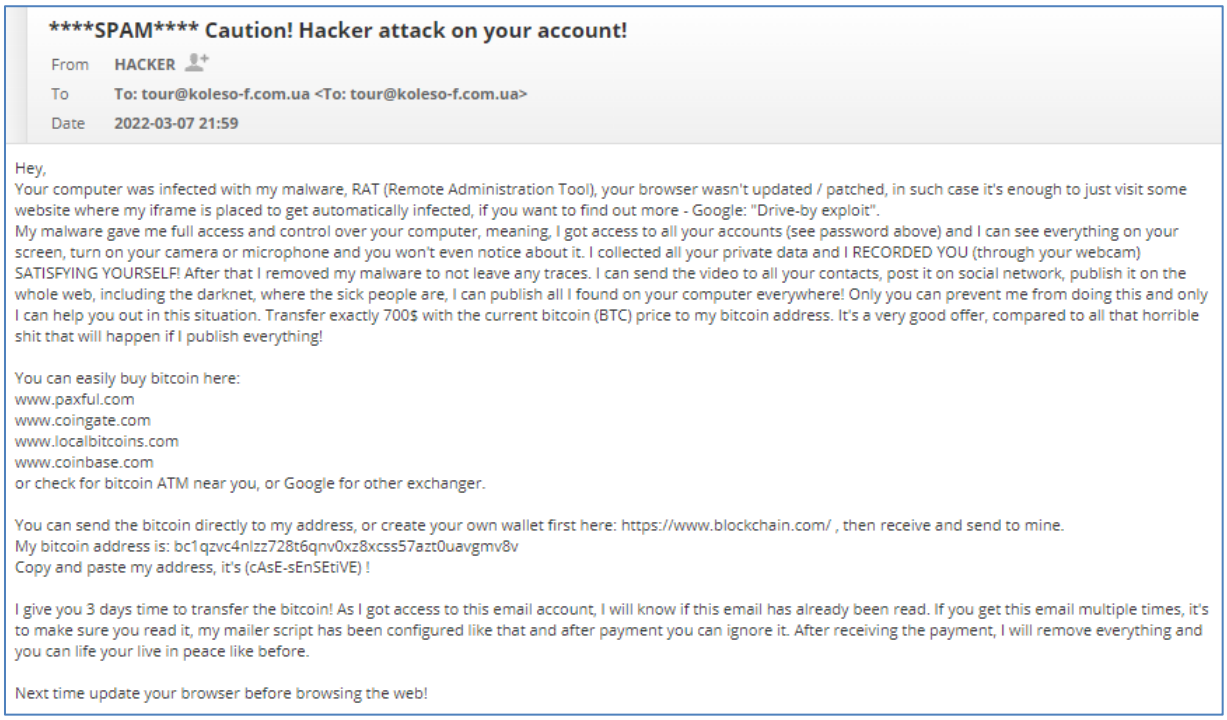


Figure 9 Sample phishing email imitating DHL

In the case of the second approach – using pressure or extortion – attackers often exercise sextortion, which usually means blackmailing the victim through claims of having

proof (video, pictures) of a sensitive nature or the victim visiting adult websites. This technique is relatively new, and the first sextortion email scams were identified only in 2018 [20].

In this scenario, the attacker doesn't have to focus on the appearance of authenticity. From the context of the email, the intent is clear, but the main focus is on the supposed leverage in the attackers' hands. Many of these email phishing attacks used leaked credentials to improve the perception of the threat. The victim might recognize the password they might have used in the past in the subject of received phishing email and believe it to be proof of a potential breach and authenticity of other claims of the perpetrator related to the potential leverage. Attackers can obtain these credentials (often gathered through various data leaks) from the dark web, where they are freely available in huge bulks or for a small fee.



******SPAM**** Caution! Hacker attack on your account!**

From HACKER
To To: tour@koleso-f.com.ua <To: tour@koleso-f.com.ua>
Date 2022-03-07 21:59

Hey,
Your computer was infected with my malware, RAT (Remote Administration Tool), your browser wasn't updated / patched, in such case it's enough to just visit some website where my iframe is placed to get automatically infected, if you want to find out more - Google: "Drive-by exploit".
My malware gave me full access and control over your computer, meaning, I got access to all your accounts (see password above) and I can see everything on your screen, turn on your camera or microphone and you won't even notice about it. I collected all your private data and I RECORDED YOU (through your webcam) SATISFYING YOURSELF! After that I removed my malware to not leave any traces. I can send the video to all your contacts, post it on social network, publish it on the whole web, including the darknet, where the sick people are, I can publish all I found on your computer everywhere! Only you can prevent me from doing this and only I can help you out in this situation. Transfer exactly 700$ with the current bitcoin (BTC) price to my bitcoin address. It's a very good offer, compared to all that horrible shit that will happen if I publish everything!

You can easily buy bitcoin here:
www.paxful.com
www.coingate.com
www.localbitcoins.com
www.coinbase.com
or check for bitcoin ATM near you, or Google for other exchanger.

You can send the bitcoin directly to my address, or create your own wallet first here: https://www.blockchain.com/ , then receive and send to mine.
My bitcoin address is: bc1qzvc4nlzz728t6qnv0xz8xcss57azt0uavgmv8v
Copy and paste my address, it's (cAsE-sEnSEtiVE) !

I give you 3 days time to transfer the bitcoin! As I got access to this email account, I will know if this email has already been read. If you get this email multiple times, it's to make sure you read it, my mailer script has been configured like that and after payment you can ignore it. After receiving the payment, I will remove everything and you can life your live in peace like before.

Next time update your browser before browsing the web!

Figure 10 Sample sextortion phishing email

1.2.2.4 Categories By Target Focus

Phishing usually refers to a non-targeted phishing attack – an attack whose target can be anyone (so-called spray and pray type of attack). Such attacks are often mass sent to a list of acquired potential victims. An example of this type of attack might be parcel delivery phishing or a sextortion scam. The usual efficacy of this type of attack is low – e.g., the average click rate (in 2021) was 17.8% [70].

On the other side stand targeted phishing attacks, where the victims are carefully selected. Also, the efficacy of this type of attack is significantly higher – the average click rate for spear phishing campaigns was 53.2%, almost three times higher than for the non-targeted attack [70].

Within this category, we further distinguish:

- **Spear phishing**
    - Targets low-profile positions with specific access, permissions, or responsibilities within the organization (accountants, IT or HR staff, etc.)
    - A widespread technique is to target newcomers in the organization as they don't usually know all the respective people in the company yet and, therefore, are more susceptible to fulfilling ad-hoc requests from their "supposed" superiors or colleagues.
    - An example of spear phishing might be previously mentioned BEC. The attacker sends phishing messages to accountants, assistants, or others within the organization. The message might be imitating a C-level manager or general manager with an urgent task or invoice to be processed.
- **Whaling**
    - It is a phishing attack against the high-ranking people within the organization.
    - An example of such an attack might be a phishing email sent to a C-level manager to acquire his credentials. If successful, these might be further used for BEC, as stated in the previous example. At the same time, the spear-phishing email to accountants will go from the actual C-level manager's email account and, therefore, be considered genuine.

Targeted phishing attacks can be tricky to spot, as they are usually prepared intently for a small group of people or even for a single person. To compose a targeted phishing message, the perpetrator usually uses personal details or information of a private nature and incorporates them into the message to improve the perception of the message's authenticity. Attackers go to such lengths that even the message delivery time is considered [40]. The easiest way to gain such specific or personal details is to turn to social networks and free OSINT tools through which the victim can be researched.

1.2.2.5 Categories By Number Of Stages

The number of stages, or intended steps through which the victim should go, is often impacted by the objective of the phishing. A single-stage attack usually gets the work done when the perpetrator intends to infect the victim's computer, e.g., the attacker could deploy a phishing email with an infected attachment. The phishing attack achieves its intended objective when the victim opens the attachment.

On the other hand, if the objective is to collect information (login credentials, payment card details, crypto-wallet details), the attacker will have to use a multi-stage phishing attack scheme. The initial phishing message represents the first stage of the attack. This stage aims to direct the victim to a pre-prepared phishing landing page or webpage, which is the second stage of the attack. The purpose of the second stage is to collect the desired information through a provided landing page form.

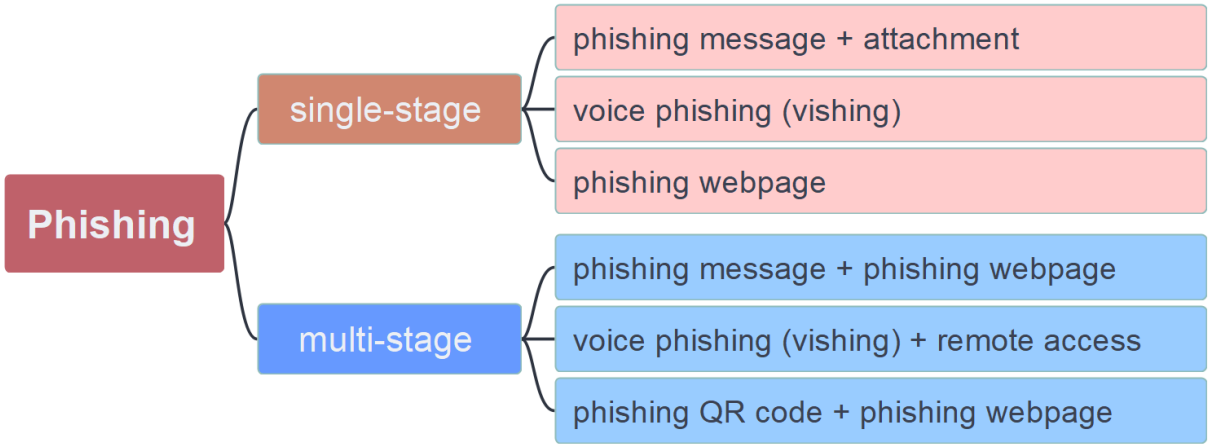


Figure 11 Phishing categories by number of stages of the attack

Alternatively, the attacker can utilize a multi-stage phishing attack to infect the victim's computer, such as deploying a phishing message that would direct the victim to a webpage or cloud storage that hosts the infected file. This attack would probably have a higher chance of success. The infected file doesn't have to pass through the email filter as an email attachment; therefore, there is a higher chance that the victim will open the infected file. Similarly, for the second scenario, the perpetrator might leverage vishing and, by using social engineering skills, might be able to acquire the information (user credentials, payment card details, or others). Nevertheless, the success factor of this approach depends on the perpetrator's persuasion and manipulation skills.

2 Distinguishing Phishing From Legitimate Webpages

As phishing attacks might follow multiple patterns, diverse detection approaches exist. For example, specific techniques, data, and characteristics would be used for phishing email detection, and others would be used for detecting smishing or vishing.

Recognizing phishing, the common characteristics and indications, and how to identify a phishing attack became a part of standard security practices related to cyber risk mitigation and awareness. These topics are included in most security frameworks and certifications (e.g., MITRE ATT&CK framework, NIST cybersecurity framework, CEH certification, OSCP certification, and many others).

2.1 How Humans Detect Phishing

Most phishing attacks, especially those targeting the general public and not a targeted one, can be spotted quickly and already in the initial stage, be it an email or other electronic communication. Numerous red flags, when observed, should trigger suspicion about the authenticity of the message or communication. These red flags are often linked to category groups. However, some indicators are shared across the different phishing typologies—e.g., suspicious origin (e.g., email, URL), pressure, and sense of urgency.

We reviewed and summarized the common indicators for phishing emails from all the channels, as they are the most commonly used channels for phishing. After the email, we will summarize the indicators for phishing webpages, as these are the prevalent choice in multi-stage attacks, and the primary focus of our research - detecting phishing webpages.

Being able to detect phishing on a webpage is an almost universal approach to mitigating the risk of phishing. Phishing webpages are used as a standalone phishing attack vector and also as part of the multi-stage phishing attack (see Figure 6). In case of a multi-stage phishing attack that started through email or any other channel (SMS, instant message, QR code, social network post, etc.), even if the initial phishing message wasn't stopped and has reached the potential victim, there is still a chance to mitigate the risk. Such mitigation could be done by analyzing the webpage accessed by the potential victim (e.g., through a web browser extension [1]). This allows for broader protection against phishing and partially eliminates the limitation of addressing each channel individually. On the other hand, this approach is

insufficient for those phishing attacks where the webpage is not being used. However, these would mostly be phishing attacks using attachments or vishing attacks, for which the prevention would primarily be awareness training. For attachments (as per [40], the three most common extensions of attachments are PDF, PNG, and JPG), it's also awareness but supported with end-point protection solutions (e.g., antivirus, hardening of the operating system, firewall, etc.).

2.1.1 Typical Signs Of Phishing Email

1. **Unusual sender** – in general, a message from this sender is very rare or was not expected at all by the recipient (bank, post office, service provider, e-shop, police, tax authority, etc.).
2. **Unusual message** – the context of the message is unexpected or unusual (e.g., request for update of credentials for Netflix or Microsoft Office, invoice for parcel victim didn't order, etc.)
3. **Email address looks suspicious** – there are multiple red flags related to the sender's email address (which might contain typos, numerical or special characters, or the domain the victim is unfamiliar with). Often, the email is spoofed and shows an actual email address, but when analyzed in detail, it shows that the email differs from the one shown in the sender field. On rare occasions, the attacker acquires access to the genuine email and sends the message from it – though this could be the actual scenario, especially in a BEC spear-phishing attack. It is also common that the person who supposedly sends the message (e.g., from the signature within the text of the email) doesn't seem to correlate with the sender's email address.
4. **Pressure and a sense of urgency** – are the most common phishing indicators. If the message wording indicates pressure or sounds urgent, it is imperative to check the other potential red flags of the message. The chances are that the message is indeed phishing. A phishing message almost always sounds urgent. To ensure the desired action is taken, it often provides leads to immediate action – sending a money transfer, validating or renewing the credentials, paying a fine or a forgotten service fee, etc.
5. **Email with attachment or URL link** – a message containing a URL link or an attachment should be considered suspicious. In the case of attachments (though less than a

quarter of the phishing emails include one [40]), the most common ones are PDFs and pictures in JPEG and PNG format. Other formats like MS Word documents, MS Excel sheets, or ZIP archives are also used frequently. Phishing messages primarily deploy a URL link [40]. The link directs the victim to the phishing landing page or malware, which would be loaded into the computer when the victim clicks it. The simplest way to mitigate the risk of falling victim is to look at the destination URL and see if there are any red flags indicating potential malicious content.

6. **Other red flags** – a generic greeting or greeting not using the receiver's name is common for automatically generated phishing emails. Poor grammar and spelling were also often observed. However, these might quickly fade away with the general availability of large language models that are capable of creating text with predefined context in a preferred language with spot-less grammar.

2.1.2 Common Signs Of Phishing Webpages

1. **Login page or payment card details form** - are the most apparent phishing page indicators. A login form is created to gather user credentials for further malicious (most common financial) gain. The second one is directly aimed toward financial theft from an unsuspecting victim.

   It is not uncommon for these entry forms to appear only after passing through the initial landing page, so in case the suspected phishing message link doesn't immediately end up in the login form or on the page where payment card details are to be filled, it doesn't mean that page is legitimate. Attackers occasionally also use multi-step phishing landing pages where the victim needs to move between phishing pages.

2. **Suspicious URL** – Whenever we are browsing the web and about to log into a webpage or provide payment details, it is best to check the authenticity of the webpage by looking at the domain or complete URL address. Some phishing attacks leverage URL obfuscation techniques to mask the URL, and attackers often register domains resembling well-known brands (typosquatting).

   Another red flag related to the URL might be that the URL seems to be a genuine webpage but unrelated to the brand the page's content is imitating – this could be a

result of a more advanced phishing attack. The perpetrator could hack into a genuine webpage web server and perform a domain or sub-domain takeover. The attacker's phishing webpages are then hosted on a safe web domain, but the owner of the domain and webpage is probably unaware of that [27]. This technique is used to mitigate the risk of the phishing webpage being marked as phishing by phishing detection algorithms that analyze the domain risk profile inside the browser.

3. **Sense of urgency** – Phishing webpages' content often works with a sense of urgency and tries to put the potential victim under time pressure. This induces stress in the victim and pushes them towards instinctive behavior rather than a cautious and logical one. This is a typical red flag for all types of phishing. In some cases, when it fits into the pretext of the message (e.g., Figure 12, which shows a phishing webpage imitating Abu Dhabi police), a sound of the siren could be played, or a countdown could be present within which the action must be taken – fine paid, credentials provided, etc.

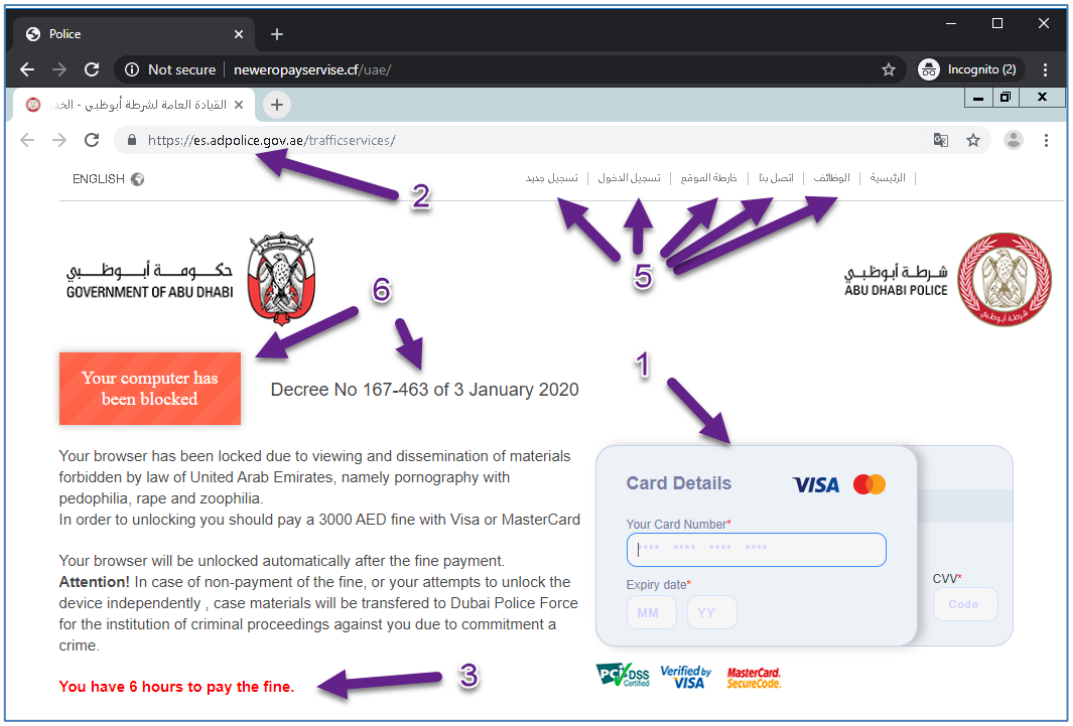


Figure 12 Phishing imitating Abu Dhabi Police with tagging of the signs of phishing

4. **Functions limiting user** – In the same pretext as the previous point (when a phishing page imitates tax authority, police, or authority capable of imposing fines or legal actions against individuals or companies), further restrictive measures could be

observed – e.g., the browser can't be closed, or resized, webpage JavaScript might be blocking the use of keyboard shortcuts or the browser menu and controls are not visible on the screen, phishing page is set as a default page for the browser, etc. The purpose of these restrictions is usually to scare potential victims and pressure them to follow the instructions provided.

5. **Missing menu or links not working** – phishing webpages try to reuse the imitated brand's design characteristics. Still, there will often be no other navigation options or a menu, or even if the menu is provided, it will just navigate back to the same phishing page. This design pattern is intended to minimize potential victims' chances of navigating away from the phishing page.
6. **Incorrect or misleading information** – Depending on the pretext, many phishing pages are built around the lie and provide inaccurate or misleading information (e.g., the provided example depicted in Figure 12 refers to a decree that doesn't exist. Another lie on the same webpage claims that the computer has been blocked, which is not true either. Still, this lie was supported by a siren sound played when the webpage was accessed, and the screen was switched to full-screen mode, resulting in hidden browser controls and a menu along with an address bar. Also, JavaScript was used to capture the keyboard shortcuts and ensure that the user cannot switch the browser back to standard view or close it. So, to the less technically savvy user, this could look like the actual work of police, especially when the computer seemed to be blocked.
7. **Other common red flags are** - the use of pop-up windows and messages, lack of registered trademarks (for well-known brands), or missing contact information on the page. Also, page design that is inconsistent with the brand used to be common among phishing web pages, though it is less common today.

2.2 How Computers Detect Phishing

While humans focus on visual clues when identifying phishing webpages, computer algorithms can leverage a variety of techniques. Simpler ones, like Blacklists or Whitelists, rely on quick and reliable data storage to archive and re-use the previous assessment status of the domain. Others – like business rules – use conditional logic to assess the webpage

characteristics and try to identify a common identifying factor for legitimate or phishing webpages to derive the final classification. Most researched techniques deploy algorithms of predictive analytics and machine learning to unravel deeper or not obvious correlations within the various observable characteristics to decide on the final result of the assessment.

2.2.1 Blacklists, Greylist, And Whitelists

The simplest solution for detecting phishing webpages is to use lists. There are two main opposing approaches whenever using list techniques – negative list (Blacklist) and positive list (Whitelist).

A Blacklist represents a list where each observation present in the list bears a negative flag or meaning. An example of such a Blacklist in our particular use case would be a list of all previously observed phishing domains collated into a domain Blacklist.

Whitelist uses the opposite concept of a Blacklist, and each observation in the list represents a positive flag; in our use case, a Whitelist could be used for collating all domains which were proven to be safe, and once the domain is whitelisted, it would be considered safe and legitimate.

In simple terms, we could deploy a list-based solution and check every domain the user is about to visit. First, whether it is present in the Whitelist (which means the domain is safe to be accessed), and then whether it is listed in the Blacklist (in case it is, the URL indeed shouldn't be visited as it was previously flagged as a phishing or malicious domain). Such a list solution could be managed on the level of a particular device (PC, mobile, etc.) where the list could be stored or as a global list centralized across the users, hosted on the publicly available infrastructure. Such a global list would require an online connection whenever it is used or updated.

2.2.1.1 Technical Feasibility Of Blacklist

From the technical feasibility perspective, there were approximately 365 million top-level domains (TLD) in the 3rd quarter of 2021 [43]. The maximum length of the domain name is 253 characters, while each label (the label is each part of the domain separated by ".") has a maximum size of 63 characters. In reality, most domains are much shorter as they try to be easy to remember. For example, the average length of the ".com" domains as per [16] is

13.539 characters, so we will round it up to 14. If we want to assign each domain to a Blacklist or a Whitelist, eventually, we would require 365 million domains * 14 (characters length, one character = 1 byte); it would be necessary to allocate approximately 5.11 GB of storage, which is certainly feasible. On the other hand, if we intended to size the storage for the maximum possible capacity, we would require 365 million domains * 253 characters in length, resulting in approximately 92.35 GB of storage.

The above calculations assume the volume of records at the SLD.TLD domain level (Figure 21). Still, further hierarchies through sub-domains (3rd, 4th, 5th level, etc.) would grow the expected size of the list significantly. Considering the average number of distinctive subdomains to be between ≈1.6 and ≈3.7 (derived as an average number of varying domains for the same registrable domain - SLD.TLD - for phishing URLs collected from PhishTank and PhishStats for the period between 2013 and 2022 depicted in Figure 13 after cleansing described in our research [IS4]) then it would be required to multiply the figures mentioned above.

As part of our research, we analyzed ten years of data from PhishStats and PhishTank where we were examining the optimal number of subdomains to be considered when building a Blacklist. It was critical to decide on the most appropriate level of granularity for domain names.

As can be seen in Figure 21, domains can have multiple sub-domains, each separated by a dot ("."), but an overall length can't exceed 253 characters (transmitted as a 255-octet packet) [55]. In our example - https://free.fr.dong.fitbet.com - the domain consists of five levels. Starting from right to left - the top-level domain "com" (TLD), followed by the "fitbet" as a second-level domain (SLD), then "dong" as a third-level domain (THLD), followed by "fr" as a fourth-level domain and concluding with "free" as a fifth-level domain.

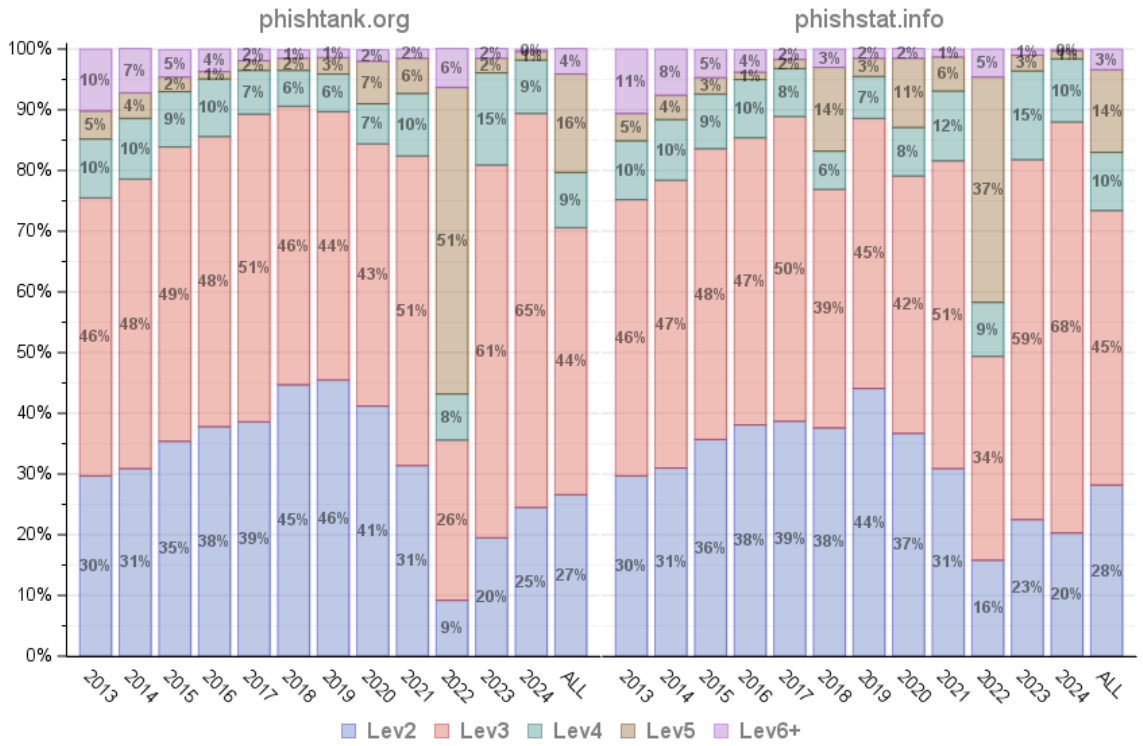


Figure 13 Ratio of subdomains count in PhishTank and PhishStats

To arrive at the most appropriate level, we calculated the prevalence of different levels of domain names in the underlying data (considering only confirmed phishing domains). In the analysis, we calculated the % share of each level of domain granularity to understand how common each level is across the analyzed period. The study was conducted on both datasets separately, and the results that were gathered can be seen in Figure 13.

Based on the data - though there are slight differences between the two datasets - initially, most domains were within 2 and 3 levels and steadily growing. Since 2019, the share of 2-level domains has started to shrink, and we have observed an increase in domains of level 4 and more. A significant shift is visible in the year (2022), where the sudden increase in domains with five levels and more is unlike any YoY change seen before. After further review of the data, we identified a sharp rise in the number of different subdomains registered to the same domain (SLD.TLD) in this period. The average number of different subdomains linked to the same domain was constantly below two throughout the whole period except the year 2022, where it almost doubled (Figure 14, an average of 3.7 different sub-domains linked to the same 2-level domain as opposed to less than 2 in the previous years). These results correspond with the findings of other researchers [56], who also observed a significant increase (+82%) in

newly registered domains in 2022 compared to 2021. We also added statistics for 2023 and 2024 to see whether this shift towards five levels continued, and it didn't. Interestingly, the share of URLs with more than four levels almost disappeared in 2023 (<4%) and also further into 2024 (<2%, though for 2024, we worked only with data from the first quarter of the year).

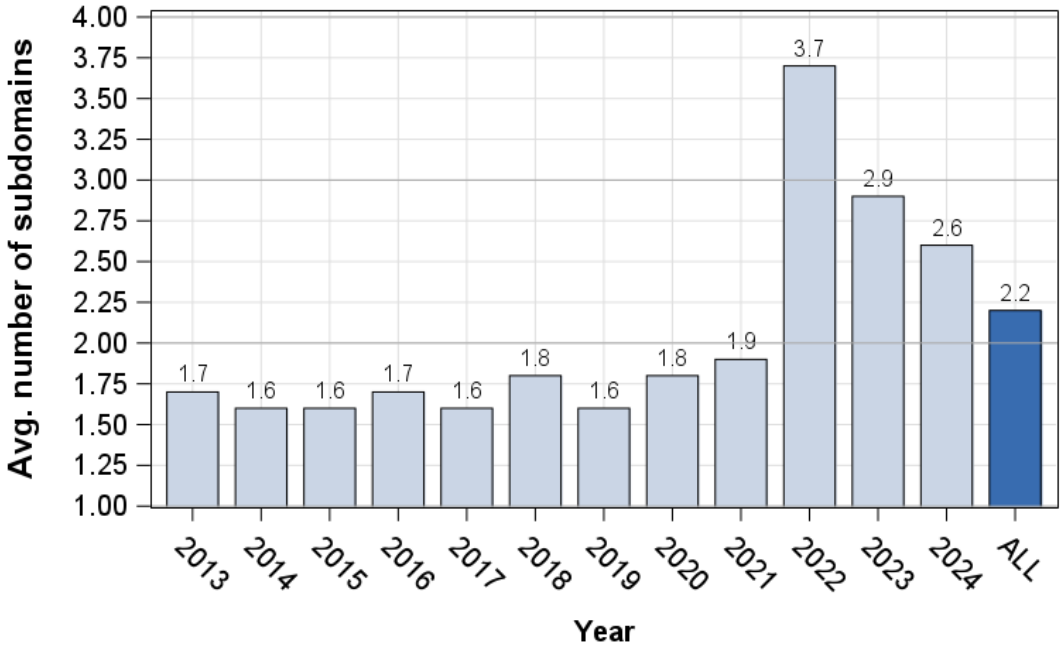


Figure 14 The average number of subdomains for PhishTank and PhishStats

As per the data, storing the domain name with a maximum of 3 levels (e.g., www.google.com) would provide only ≈71% and ≈73% accuracy (PhishTank and PhishStats data across the whole 10-year period – depicted as "ALL" in Figure 13). More recent data (from 2019 to 2021) show an increased share for domains with 4 and 5 levels. The final recommendation is to proceed with the domain names of phishing URLs with five levels of accuracy to keep the accuracy above the ≈90% mark (≈96% for PhishTank and ≈97% for PhishStats).

2.2.1.2 Domain Blacklist

Domain Blacklist was among the first detection techniques used against phishing, usually as part of the web browser. This is still true today, as all commonly used modern browsers carry a highly accurate phishing detection functionality [51][52]. Already in late 2004, criminal groups were focusing on phishing attacks like the known "Rock Phish" group, which employed

single-use URLs. This approach caused concern among security professionals as it bypassed most existing anti-phishing solutions that relied on URL lists at the time[53].

Blacklist is built chronologically as data are captured in the real world. Only confirmed phishing records (True Positive - TP) are added to the list. Blacklist captures the domain and date when it was classified as confirmed phishing (TP). For practical reasons, as discussed in the limitations of the Blacklist-based approach, it is desirable to build a Greylist that would capture reported domains classified as non-phishing (FP). Greylist is named in such a way because it does not capture only legitimate pages, but it could also be used for domains whose classification score is low, and the final class might not be accurate. Such a Greylist can be used to contain domains that were reported as legitimate and phishing at some point in time; therefore, their classification is ambiguous (e.g., some well-known legitimate domain becomes a victim of a hack or sub-domain takeover).

The process for a newly reported domain passing through a Blacklist-based solution is depicted in Figure 15. Every record first passes through the **Assessment** steps - checking whether the domain exists in Greylist or Blacklist. If the domain is found in Greylist, this record is flagged as ambiguous (UNK), and processing is ended. If the domain is found on Blacklist, the record is flagged as confirmed phishing (TP) and processing is ended. Suppose the record wasn't found in any of the lists. In that case, it continues with the **List update** step, where the classification algorithm provides the assessment of whether the record is phishing or a legitimate webpage. If the record is flagged as a legitimate page, its record is added to the Greylist. If the record is classified as phishing, it is placed on the Blacklist.

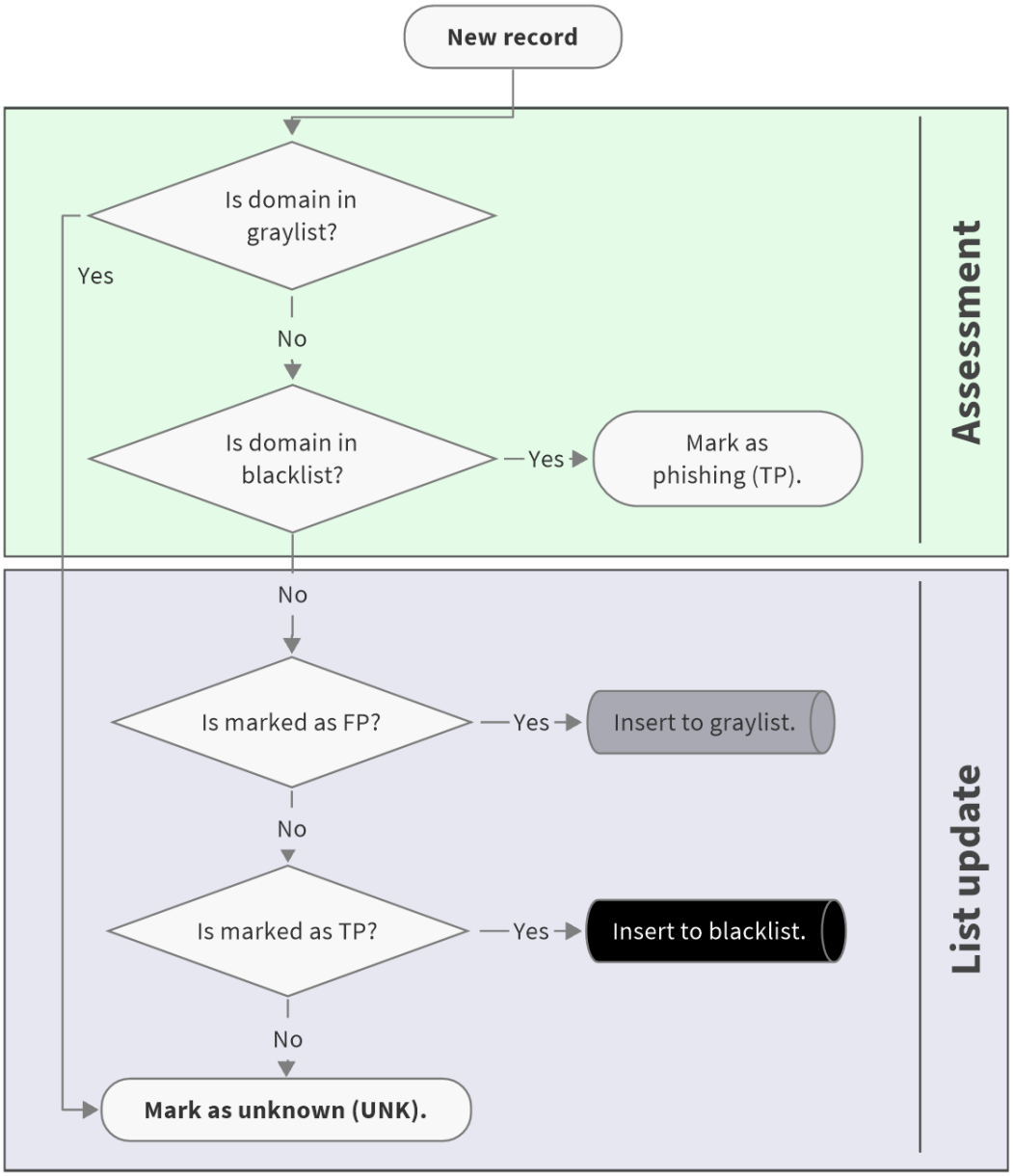


Figure 15 Flow of a new record passing through the Blacklist-based solution

2.2.1.3 Limitations Of Blacklist

Domain Blacklist has two inherent characteristics which limit its use or efficacy:

- it can't assess domains that were not previously classified [39]
- it requires another classification technique to support updating the list

The first point directly impacts the efficacy of the domain Blacklist. If a ratio of re-occurring phishing domains can be identified, it would be possible to formulate the theoretical maximum efficacy of a domain Blacklist. If only a fraction of domains is re-occurring, then only

this fraction of domains can be effectively assessed against the Blacklist. As per our empirical analysis, considering all the data across a ten-year period, 78% of confirmed phishing attacks were hosted on unique domains, and less than 22% were hosted on re-occurring domains (Figure 16).

	Year										
	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	All
RE-OCC	21.5%	27.5%	35.5%	38.4%	37.4%	33.5%	32.1%	37.8%	14.1%	6.1%	21.8%
UNIQUE	78.5%	72.5%	64.5%	61.6%	62.6%	66.5%	67.9%	62.2%	85.9%	93.9%	78.2%

Figure 16 YoY % share of re-occurring vs. unique phishing domains

The year-over-year trend view initially shows an increasing share of reoccurring domains between 2013 and 2016, from 21.5% to 38.4%. In the more recent period - from 2020 to 2022 - we see a sharp decrease from 37.8% down to only 6.1%.

From domains that re-occurred, the vast majority did so only once (almost 68%), some were re-used twice (less than 17%), and only a few were re-used three(6%), four(3%) or five times(less than 2%) as depicted in Figure 17.

Re-occurence	
1	67.8%
2	16.8%
3	6.5%
4	3.3%
5	1.9%
6	1.2%
7	0.7%
8	0.5%
9+	1.3%

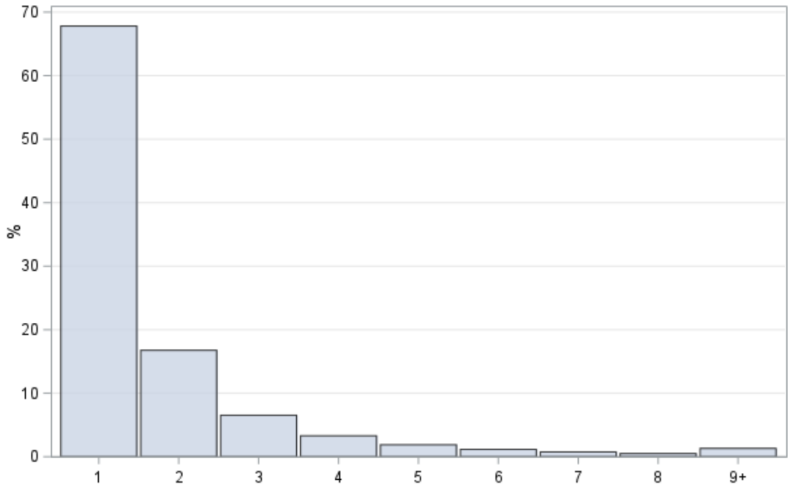


Figure 17 Frequency of the phishing domain re-occurrence

The second point - a need for a new domain classification process or technique - is critical for updating the list. Domain Blacklist needs to store previously seen phishing domains. There are two central assessment approaches

- human-driven
- automated/machine-driven

The human-driven approach requires a human to decide whether or not to flag the domain as phishing. The machine-driven approach most commonly leverages classification machine learning algorithms [36],[54].

Another inherent limitation of the Blacklist approach is that it works with the premise that once the domain is marked as good(Whitelist) or bad(Blacklist), it will remain so forever, which is a wrong assumption. The domains can be bought or sold, but significant risk comes from the fact that domains can be hacked or forcefully temporarily taken over. Suppose a legitimate domain can be misused (even accidentally) and host a phishing page. Putting the domain on a Blacklist could cause a severe business impact (imagine a domain like facebook.com or outlook.office365.com marked – even temporarily – as a phishing page due to hacking). Such an attack might also be made to cause a business impact on a particular company via "poisoning" and flagging a genuine domain or IP as malicious and letting their protective solutions cause a business interruption. Therefore, as stated above, adding a domain to the list would require a very accurate classification technique.

2.2.2 Rule-based Techniques

Rule-based techniques operate on a set of predefined criteria or patterns known to be indicative of phishing attempts or the opposite. These rules can be derived from various characteristics of webpages, including URL structure, webpage content, and domain information. Unlike AI/ML techniques that require training on datasets, rule-based systems rely on the logical application of these rules to assess webpages in real-time.

Rule-based techniques are usually represented via commonly known patterns, which are strong phishing indicators. The most common use case for rules-based techniques is the presence of known URL obfuscation techniques. As part of our research, we presented an analysis of the prevalence of common URL-based obfuscation techniques on ten years of phishing data [IS5]. Other common patterns might be related to apparent discrepancies – e.g., a webpage imitating a known brand hosted on an unusual or unknown domain. The presence of typos and mistakes in the presented language is especially indicative of the local languages of smaller countries. An example of rules identifying legitimate URLs could be IP ranges dedicated for local networks (e.g., Class A 10-dot range, Class B 172-dot range, or Class C 192-

dot range) as these are not routable on the internet, and if typed into the browser, usually mean a resource available on the local network.

2.2.2.1 URL Obfuscation Techniques

The use of obfuscation techniques on URLs or domains to commit a scam is a form of semantic attack [59]. Semantic is a study of meaning and symbolization, and therefore, such attack focuses on the users and their interaction or interpretation of what they observe, ultimately exploiting the human factor. In [57], the author describes a specific obfuscation via the use of the at sign (@) as an attack on the user's preconceived notion about the meaning of a URL. Richard Siedzik [58] describes semantic attacks targeting human elements or human nature through which people assign meaning to content. This type of attack is based on the knowledge that most internet users don't know what a URL should look like and what components and structure it holds.

Examples of other types of obfuscation techniques that are commonly used but can't be identified from the domain or URL are redirects (deployed on the client side via <meta> tag forcing refresh, javascript, or deployed on the server side); another example is QR codes.

Obfuscation techniques can serve diverse objectives, but the two most important ones are **evading detection** and/or **increasing credibility**, which are often coupled. In some scenarios, obfuscation techniques can improve on both; in others, they might counteract. Puny code is an example of an obfuscation that positively impacts both objectives. The potential victim sees and might believe to be accessing the genuine domain. The chances of phishing detection are significantly reduced because one of the most common clues - domain or URL perception was passed, and the credibility of the currently visited domain increased. When improving one objective reduces the other - an example of the opposite scenario is replacing the domain name with an IP address. Using an IP address could help the attacker bypass the domain watch lists, but it might reduce credibility in the eyes of the victim when the URL is shown with IP in the browser's address bar. To counterbalance this negative impact, the attacker might deploy another technique - secured HTTP (HTTPS) to improve the site's credibility in the eyes of the user. The most common obfuscation techniques used are:

- Obfuscation using the at "@" sign – "@" sign has a specific purpose in the URI as part of the authority component; see Figure 21. Part preceding the at "@" sign is a user information sub-component, which is used only rarely (due to security reasons - passing cleartext credentials [57]). Nevertheless, using this sub-component can help the attacker to deceive the potential victim. An example of such an attack is **http://dhl.cz:0@www.dongfengcidef.cl**, which tries to evoke the visiting **dhl.cz** domain, while in reality, the browser will navigate to a webpage hosted on **dongfengcidef.cl** domain.
- Obfuscation via HTML entities – HTML entities are easy to identify as they always begin with an ampersand "**&**" and end with a semicolon "**;**". There are two types:
    - Named HTML entities - are most commonly used to display characters with special meaning in HTML, like less-than sign "**<**" written as "**<**" used for the HTML tag opening or greater-than sign "**>**" written as "**>**" used for closing the HTML tag.
    - Numeric HTML entities - which are used to express any character using the hexadecimal ("**&#xHH;**") or decimal format ("**&#DD;**"). For example character "**@**" can be expressed as "**@**" or "**@**".
- Obfuscation by specifying port details – To make malicious URLs more convincing, attackers can use obfuscation techniques by explicitly mentioning the port number right after the colon character "**:**" placed at the end of the host component (e.g., **http://google.com:80**), see Figure 21. Another intent might be to make the URL look more complex and focus the user's attention on the port part of the domain while ignoring the preceding domain part, which points to a malicious site. The last use case is targeting a firewall, which might be configured to filter out traffic passing through specific ports. Attackers can leverage non-standard ports to bypass such firewall rules. In some cases, the port colon was present, and the actual port number was omitted. There were only single-digit occurrences each year for such cases.
- Use of Punycode to mimic genuine domains – Punycode is an encoding of a non-ASCII Unicode string into an ASCII string. It was defined in 2003 in RFC3492 [60]. The presence of Punycode can be identified through "**xn--**" prefix within the string. Intended regular use of the Punycode allows users to type a domain name into the

browser's address bar in their language-specific character set like Chinese, Cyrillic, and others. A Unicode string is translated using the Punycode encoding algorithm within the browser into an ASCII-compatible string, which is then sent to DNS to return the IP address of the requested domain.

Punycode can be highly efficient for homograph attacks or brand spoofing by replacing certain ASCII characters in the domain with a non-ASCII Unicode character, which looks identical or very similar to actual ASCII characters. For example, the URL **http://account.xn--googe-wsa.com/** is presented as **http://account.googìe.com**, another example **http://app.xn--sshi-08a.tk/** is shown as **http://app.sụshi.tk**. The examples show that these character replacements are hard to spot, especially if the characters are carefully selected.

- Obfuscation through IP address – Substituting the domain name with an IP address in the URL of a phishing web page is the most prevalent technique of URL obfuscation. The most common objective of such substitution is hiding the actual domain name - which might expose the phishing nature of the webpage to the potential victim. IP addresses can be represented in various notations:
    - **IPv4 notation** - the most commonly used and known xxx.xxx.xxx.xxx where xxx is a number between 0-255, e.g., http://211.72.122.11/secured/index.htm
    - **Single value notation** - IP is represented as a single value ranging from 0 to 2^{32}, e.g., http://1077629123/phpma/config/ (in IPv4: 64.59.80.195).
    - **Hybrid notation** - IP is represented as a variation of the above two techniques, e.g., http://0x4a.0x361142/~cgipecom/www.irs.gov (which can be represented as http://74.3543362/ by converting hexadecimal values into decimal and which further translates into 74.54.17.66 in IPv4)

  IP written in the above notations can also represent the numerical value in different formats. The most common are:
    - **Decimal** - IPv4 notation example: http://66.147.240.156/~frpaypal/, single value notation example: http://1075516530:82/index.php and hybrid notation example: http://203.10654640:8080/.https/www.wellsfargo.com
    - **Hexadecimal** - can be identified through the specific prefix "**0x**". IPv4 notation example: http://0xd8.0xb6.0x6c.0x58/signin/, single value notation example:

http://0xd2bb6e92/.b.php and hybrid notation example: http://0xa8.0xbb5ce5/vsp/form.html

  - **Octal** - can be identified through a leading zero character "**0**". IPv4 notation example: http://0106.0125.0326.0102/www.poste.it/login.html, single value notation example: http://033113520761/start.jsp.htm and hybrid notation example: http://0125.027135477/aw/
  - **Combined** - combines the above numerical formats, e.g., http://0x6b.026.0320.189/, which combines hexadecimal with two octal and one decimal format within the IPv4 notation.

- URL shorteners – URL shorteners were designed for convenience to simplify the sharing of longer URLs, but malicious actors started exploiting them to obfuscate phishing URLs. URL shorteners substitute a URL with a short hash code right after the link to the shortener's primary domain, e.g., **http://bit.ly/13mod8** or **http://tinyurl.com/ykplrqz**. There are hundreds of URL shorteners today (in our analysis, we identified more than 250).
- Employing HTTPS to appear legitimate – The idea behind using HTTPS on phishing sites is to make them seem more legitimate in the eyes of the potential victim. By configuring HTTPS on the server side, the visitor's communication between the local device (PC, mobile, etc.) and the server becomes encrypted instead of only HTTP cleartext communication, which can be eavesdropped on. Practically, HTTPS has no relevance regarding the potential phishing purpose of the hosted site or provides no risk mitigation.

  As per Figure 18, the shift towards HTTPS is obvious and confirms what APWG presented in a report from Q3/2020 onwards: More than 80% of phishing pages were already set up with HTTPS. Our numbers show slightly lower figures—the most recent data in 2023 shows ≈72% among confirmed and ≈78% among unconfirmed phishing records.

	Source data			
	Phishing		Unconfirmed	
	Share %		Share %	
	http	https	http	https
Year				
2009	99.9%	0.1%	99.6%	0.4%
2010	99.8%	0.2%	99.1%	0.9%
2011	99.6%	0.4%	98.7%	1.3%
2012	99.4%	0.6%	98.5%	1.5%
2013	99.5%	0.5%	98.5%	1.5%
2014	99.3%	0.7%	98.1%	1.9%
2015	99.1%	0.9%	97.7%	2.3%
2016	98.2%	1.8%	96.0%	4.0%
2017	87.6%	12.4%	82.9%	17.1%
2018	71.7%	28.3%	66.9%	33.1%
2019	54.2%	45.8%	45.4%	54.6%
2020	47.9%	52.1%	39.3%	60.7%
2021	41.2%	58.8%	30.7%	69.3%
2022	59.3%	40.7%	27.1%	72.9%
2023	28.1%	71.9%	20.3%	79.7%

Figure 18 Prevalence of HTTPS in PhishTank and PhishStats between 2009 and 2023

Though obfuscation techniques are commonly used as a strong indicator of phishing (sometimes to the point where their presence is considered a confirmation of a webpage being phishing), this is not always the case, and it is advised to use such generalizations cautiously. Each listed obfuscation technique has a legitimate use case, though the general experience leans towards considering such occurrences as phishing indicators.

2.2.2.2 Benefits And Limitations

The benefit of this approach is that it's possible to classify or tag never-before-seen domains based on formulated rules (a weakness of the list-based solution). Also, rules are more efficient from a required storage perspective, as no significant storage is needed (compared to the list-based technique). On the negative side – this approach requires a periodic re-evaluation of the deployed rules; otherwise, the classification accuracy might deteriorate over time (an example would be using the HTTPS vs. HTTP prevalence).

2.2.3 Algorithms Of Machine Learning

Data analytics, especially machine learning, have become trendy in computer science in the last couple of years, primarily because of the potential to change the decision-making process within the organization across industries.

Analytics definition by company SAS:

> *"Analytics uses **data** and **math** to answer **business questions, discover relationships, predict unknown outcomes,** and **automate decisions**. This diverse field of computer science is used to find meaningful patterns in data and uncover new knowledge based on applied mathematics, statistics, predictive modeling, and machine learning techniques."* [38]

2.2.3.1 Types Of Data Analytics

Data analytics encompasses a range of techniques and methodologies used to extract insights and information from data. Depending on the questions, we can apply different types of analytics to transform the raw data into actionable intelligence. These types range from basic descriptions of past events to advanced predictions and decision-making strategies for future actions. Below, we explore the four main types of data analytics, each characterized by its unique approach and utility in organizational contexts:

- **Descriptive**
    - Is answering the question - **What and how?**
    - Is represented by reporting and business intelligence tools.
    - Examples of descriptive analytics are static or dynamic reports.
- **Diagnostic**
    - Is answering the question - **Why?**
    - Is represented by a business intelligence tool that allows exploratory analysis.
    - An example would be queries and drill-downs to identify the source of unusual observations in the data.
- **Predictive**
    - Is answering the question - **What will happen if?**

- Is represented by predictive and statistical models applied on top of the data and helps to identify relationships and trends within the data.
- An example could be a classification model recognizing phishing emails.

- **Prescriptive**
  - Is answering the question – **What is the best action if?**
  - Is represented through mathematical and optimization models.
  - An example could be a model calculating the best combination of raw materials considering different conditions – price, distance, quality, etc.

Companies can utilize all of the above-listed types of analytics simultaneously. Still, not every company can use any of the above techniques, as they are usually deployed in a staggered manner depending on the maturity of the given organization and the the questions the organization needs answered.

2.2.3.2 Analytics Maturity Ascendancy Model

Organizations that want to deploy the analytics techniques successfully have to fulfill different requirements directly linked to the digital transformation of the business operations - an ability to collect and distribute the data to consumers across the company on time and consistently. Because of these dependencies, the analytics maturity ascendency model (prepared by Gartner [11]) links each analytics technique's benefits to the underlying requirements' difficulty and has ordered the above four types in a sequence. Descriptive analytics and its techniques are the ones the organizations should start with, seamlessly advancing toward diagnostic analytics. These two techniques utilize historical data and answer questions related to past events. The following two types - predictive and prescriptive analytics - focus on the future using historical data.

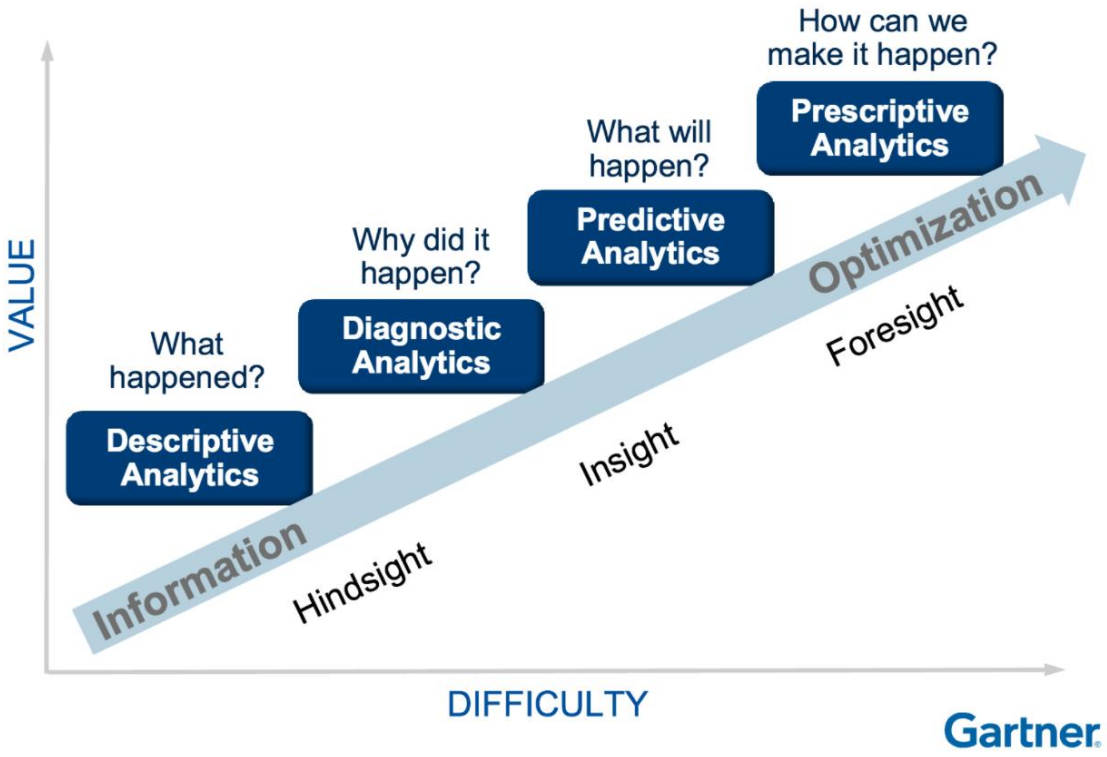


Figure 19 Gartner's analytics maturity ascendancy model [11]

2.2.3.3 Machine Learning

As the name indicates, machine learning is about enabling computers to learn (most commonly through a big enough sample of data) a particular task independently, without requiring a programmer to construct the logic on how to perform the given task or operation. This is possible through different algorithms available under the Machine Learning domain.

Tasks within the realm of descriptive and diagnostic analytics usually don't require or apply machine learning algorithms; instead, they leverage data analysis tools that allow data profiling, aggregations, and statistical calculations. Predictive and prescriptive analytics heavily rely on machine learning algorithms and deep learning. An example of machine learning could be identifying fraudulent credit card transactions, predicting customer churn, or identifying the best next offer for a given customer based on his characteristics and previous purchases.

There are a couple of categories of machine learning algorithms. Machine Learning algorithms are bifurcated based on the existence of the target variable among the historical data, which is used to derive the model (e.g., in our research - an indicator that the webpage

is phishing or not). Algorithms using the target variable belong to the group of **supervised learning** algorithms [30]. Those not requiring the target variable are grouped into **unsupervised learning** algorithms.

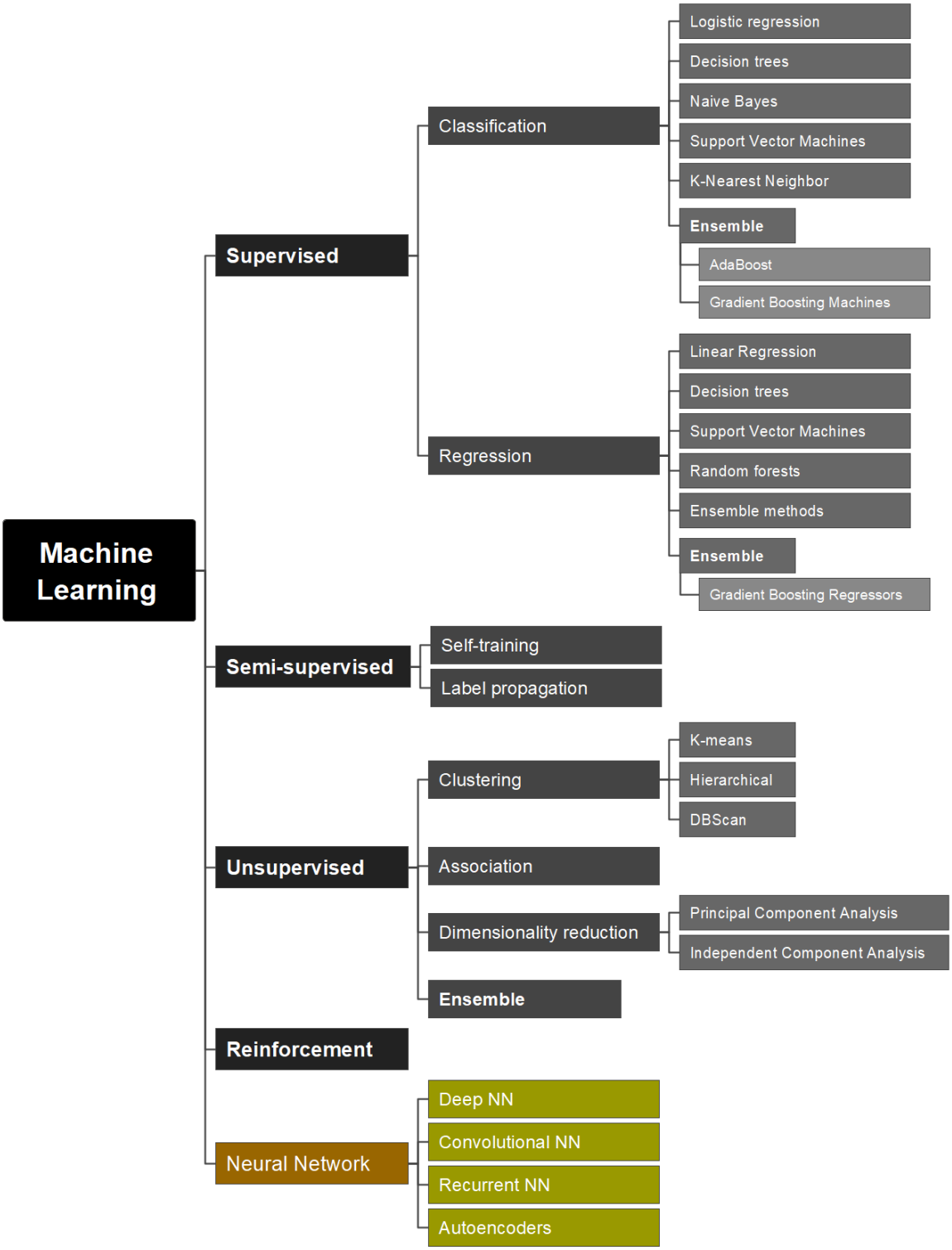


Figure 20 Categories and types of the most common Machine Learning algorithms

For phishing webpage detection purposes, we will focus on supervised learning algorithms and, more specifically, on the subgroup of **classification algorithms** where the expected outcome is a discreet class label (in our case, the decision of whether a webpage is phishing or not). Another sub-group of supervised learning algorithms is represented through **regression algorithms** (see Figure 20). These work well with continuous target variables (an example of a use-case could be a car price estimation based on mileage, brand, age, etc.). **Neural networks** are also heavily used in classification and regression tasks.

Overview of applications of machine learning algorithms:

Authors in [42] applied a logistic regression algorithm to assess the URLs of the domain and classify them as phishing or legitimate. They used a technique called bag of words (BoW) to break down the domain part of the URL, along with some modifications to improve the accuracy of the derived model. The final accuracy achieved through multiple variants ranged from almost 95% to 97%.

Jain and Gupta [73] analyzed the hyperlinks (URLs) and the webpage's content and created 12 different characteristics to train various models. The logistic regression algorithm achieved the highest accuracy of 98.42%, followed tightly by the random forest algorithm with an accuracy of 97.37%.

Sameen et al. [74] have achieved similar results with their PhishHaven algorithm, where the true-positive ratio for logistic regression was among the highest (96.71%) across a variety of algorithms (SVM, Neural Networks, K-Nearest Neighbor, Gradient Boosting, etc.). In their experiments, the Decision Tree and Random Forest algorithms fared a bit better (96.75% for both).

A model assessing a webpage purely on its URL is very practical and fast, as all that is needed is the URL to identify whether it is a phishing attempt. Such an approach was followed by Sahingoz et al. **Error! Reference source not found.** and also Wang et al. [46]. Sahingoz [61] applied seven various classification algorithms. Random Forest achieved the highest accuracy at 97.98%, and the second highest accuracy was achieved by the Decision Tree algorithm at 97.02%.

Wang's [46] proposed PDRCNN algorithm combined Convolutional and Recurrent Neural Networks. A recurrent neural network was used to extract the global features, and a

convolutional neural network was used to extract local features. Wang performed a variety of experiments and achieved an accuracy of 95.6% with a very robust model tested on phishing data from 2007 till 2018.

Kumar et al. [15] enriched the usual URL-based features with domain details from the Whois database, such as domain age, months to expiry, and zip code of the domain holder. Similarly to outcomes achieved by other researchers described prior, the best accuracy was given by random forest and decision tree (98.03% and 98.02%) followed by the K-nearest neighbor (97.99%), Logistic regression (97.7%) and finale Naïve Bayes (97.18%). Accuracy results are placed tightly within a 0.85% range. Though we were focusing on accuracy as an indicative measure of the efficacy of the applied algorithm across various experiments, Kumar et al. [15] also provided the ROC curve along with the Area Under the Curve (AUC) metric for all tested algorithms, which helps objectively compare the performance of different models. A higher AUC value indicates a model with better performance in distinguishing between phishing and genuine webpages. Reviewing the AUC from the experiments, it was Naïve Bayes that had the highest value (0.991), followed by Decision Tree (0.989) and K-Nearest Neighbor (0.987). So, based on this metric, the most accurate model was trained using the Naïve Bayes algorithm, though the accuracy metric rated it as the least accurate.

Kulkarni and Brown [29] evaluated four algorithms – the Decision Tree, Naïve Bayesian, and Support Vector Machines and added a Neural network with three layers and backpropagation. The difference in their approach is that they classify the URL into three classes: phishing, suspicious, and legitimate. Practically – classifying input into a “Suspicious” class does not provide value to the end-user as the solution should clearly state the outcome of assessment as “safe” for a legitimate page and “unsafe” for phishing or suspicious pages. Nevertheless, such classification might be relevant as a pre-assessment, deciding whether a specific model should be used for ambiguous(suspicious) or hard-to-decide webpages or URLs. Overall accuracy measures are lower than in the previously mentioned works of other authors – the highest accuracy was achieved by the Decision Tree algorithm (91.5%), followed by the Support Vector Machines (86.69%). The third was the Naïve Bayes algorithm (86.14%), and the least accurate was Neural Network (84,87%). Lower accuracy measures might be the result of assessing into three classes (phishing, suspicious, and legitimate) as opposed to the usual

two (phishing vs. legitimate) but also as a result of the selected dataset, which used only a few hundred records (702 phishing records, 548 legitimate records and 102 suspicious).

The same dataset was used by Waleed and Ahmed [3] to evaluate the accuracy of the Deep Neural Network (DNN) and Back-Propagation Neural Network (BPNN) compared to the set of other classification algorithms. Based on the experiments, the highest accuracy was achieved by DNN (88.77%), followed by BPNN (87.14%). The remaining algorithms achieved lower accuracy figures – KNN (87.07%), Decision Tree (84.9%), Naïve Bayes (82.11%) and SVM (80.78%).

Lokesh and BoreGowda [25] have used one of the commonly used datasets with 30 characteristics, containing ≈6000 phishing and ≈5000 legitimate pages collected in 2015. They evaluated a set of algorithms from which the highest accuracy was achieved by Random Forest (96.87%) followed by the Decision Tree (96.05%). The third place belonged to the K-Nearest Neighbor (93.53%), and the fourth was the Linear Support Vector Classifier (92.69%). Surprisingly low accuracy, 48.56%, was achieved by One class Support Vector Machine algorithm. Random Forest also had the highest accuracy (96.9%) in the work of Zamir et al. [48], followed by the Neural Network (95.8%), K-Nearest Neighbor (94.2%), and Support Vector Machines algorithm (93.1%). The lowest accuracy was achieved by Naive Bayes (72.67%). It is important to note that this work used the same dataset as [25]. So, the best accuracy achieved by Random Forest is not surprising.

Babagoli et al., in their research [8], employed the Support Vector Machine algorithm and the Decision Tree algorithm used for feature pruning. The accuracy of detection was 92.6% on training data and 91.8% on testing data.

Neural Networks constitute a separate subset of the machine learning algorithms. They differ from previously described machine learning methods due to their versatility and practical applicability across categories – they are efficiently used within the supervised, semi-supervised, unsupervised, and reinforcement categories. Neural network algorithms can extract the features instead of the other classification algorithms, where the derived characteristics (features) must be prepared. Mohammad et al. [76] proposed and tested a neural network algorithm that incorporated an automated pre-process to evaluate the best

architecture of the neural network (layers and neurons). Their algorithm achieved an accuracy of 91.31%.

Yang, Zhao, and Zeng [47] proposed MFPD algorithm that examines the characters in the URL of a website. It uses a combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs) to analyze these characters for patterns typical of phishing sites. CNNs help identify local patterns or features in the sequence of characters in the URL. At the same time, LSTMs assess the order or sequence of these characters, picking up dependencies that might indicate a phishing attempt. After the initial analysis, the algorithm incorporates statistical features from the URL, the website's code, and the text on the webpage. This multi-dimensional analysis provides a robust basis for detection compared to looking at just one type of data. Validation of the trained model showed accuracy at 98.99% and FPR at 0.59%, both very high numbers.

Machine learning models have limited capacity to observe new patterns not present in the underlying data. Another important aspect of machine learning algorithms is their need for model re-training due to changes in the structure or values within input data. Consider the fact that attackers are continuously trying to counter the newest anti-phishing techniques and identify new "loopholes" to exploit and increase the efficacy of the phishing attack (e.g., deploying HTTPS protocol instead of HTTP to strengthen the perception of security from the potential victim's perspective). Phishing is ever-evolving and adaptable, and the characteristics of phishing web pages reflect these changes over time; therefore, the model's accuracy decreases. After updating the training dataset with fresh data and re-training, the model will reflect the recent trends more accurately. This repetitive process of incorporating recent observations as a feedback loop creates a requirement and dependency on continuous data collection and storage.

2.2.3.4 Layered Or Hybrid Combined Approach

Current research in phishing detection leans more towards techniques like predictive analytics and machine learning, which were proven to be highly accurate, with accuracy ranging from 84% up to almost 99%, while the majority of the assessments claimed accuracy above 95% [27] and unlike domain Blacklist can also assess never before seen domains; however, supplementing these techniques with a Blacklist to achieve even a marginal gain

would practically be translated into significant financial as well as non-financial savings due to number of the impacted victims globally.

As described in Chapter 1.2, phishing is a multi-faceted problem, and even though researchers claim that their method is highly accurate (accuracy above 95%), the devil is hidden in the details. They rarely apply comprehensive testing data with very diverse phishing typologies from a more extended period to validate their results. Therefore, many of the tested techniques claim very interesting KPIs but would probably face much harder scrutiny when deployed into a real-world solution.

Building a real-world solution to detect phishing would require a layered or hybrid approach, combining multiple techniques. One such approach is presented by Rahman et al. [32] by deploying a set of machine learning algorithms – Random Forest, Decision Tree, Multi-Layer Perceptron, Support Vector Machine, Stochastic Gradient Descent, and Gaussian Naïve-Bayes. These were used in the first level as a base classifier of stacked generalization. An XGBoost classifier was used to make the final prediction. The stacking concept was implemented by applying 10-fold cross-validation in the first level. As the algorithm was applied to 3 distinctive datasets, the resulting accuracy ranged from 96,8% to 97.9%. Another example of a hybrid approach was proposed by Abdelhamid et al. [75]. They combined association rules (if-then formatted rules) with a classification machine learning algorithm, achieving an accuracy of almost 95% even after reducing features from 16 to 9 based on the correlation metrics.

2.2.4 Phishing Webpage Indicators (Characteristics)

Phishing webpage indicators or characteristics are various properties of the webpage that can be used to distinguish a phishing webpage from a legitimate one. The main goal of preparing the phishing characteristics (feature engineering) is to identify and prepare features that most effectively distinguish between the classes we want to predict – in our case, differentiate between phishing and legitimate pages. Therefore, the best features are those that make it easier for the model to distinguish between the classes. The best features should capture the "signal" — the underlying patterns relevant to class distinctions — and minimize "noise" because irrelevant data could confuse the model. By selecting the right features, we can simplify the model. A simpler model is generally easier to interpret, faster to train, and

less prone to overfitting than models that must handle many irrelevant or less informative features. Robust features help the model perform well not only on the training data but also on unseen data (generalization), making the model more reliable and practical for real-world applications. Incorporating expert phishing domain knowledge can guide the feature engineering process to focus on variables known to be class distinctions' determinants.

2.2.4.1 Phishing Indicators – URL

The most commonly used features are derived from webpage URLs. As stated in the previous chapter, deciding whether the URL is legitimate or a potential phishing attack solely based on the URL has a huge advantage. This assessment can be swift as it requires only immediately available information—the URL itself.

Characteristics are derived from the whole URL or its logical components, as depicted in Figure 21. Frequently used ones are related to length, number of specific characters or words, and presence of special characters or obfuscation techniques (2.2.2.1 URL Obfuscation).

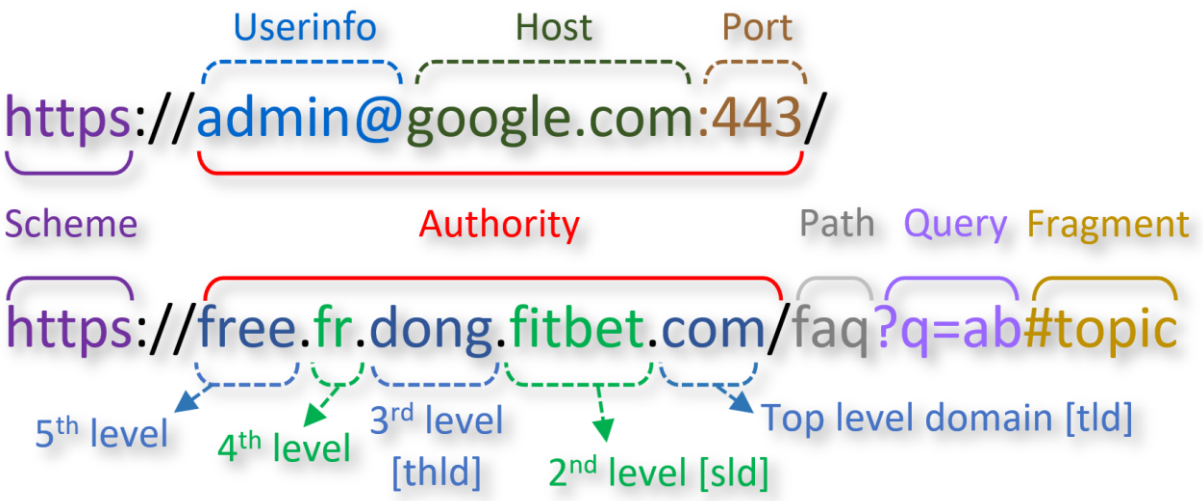


Figure 21 URL syntax diagram

As part of our research, we consolidated a comprehensive list of 155 URL-based features (see ***Appendix 1 – URL-based Characteristics***). These features were derived from the URL itself or the following sub-components:

- URL – complete URL with all components together with 26 characteristics
- Scheme – 8 characteristics
- Authority – 48 characteristics

- Path (folders path + filename) – 23 characteristics linked to the folder's path and 25 related to the filename
- Query – 20 characteristics
- Fragment – 5 characteristics

Characteristics were represented as number - counts, flags (0 or 1), or percentual share. This list is not exhaustive and can be further extended with additional features by separating the textual and numerical parts, applying various tokenization approaches, or clustering high-prevalence phishing characteristics (e.g., list of the top 10% of TLDs observed among phishing URLs but removing 10% of most common TLDs among legitimate URLs).

2.2.4.2 Phishing Indicators – Webpage

This approach considers different objects related to the phishing web page beyond just the domain or URL (**Figure 21**), deriving the indicators from the potential phishing web page's content. It uses characteristics linked to the objects on the page that might be perceivable from the user's perspective or hidden in the webpage code.

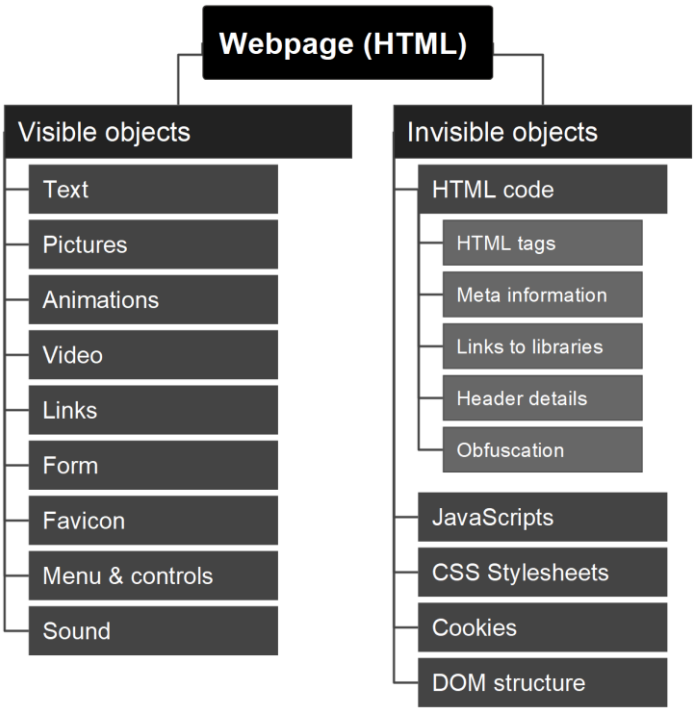


Figure 22 HTML webpage objects

Many phishing webpages try to collect credentials by imitating login webpages for common services. Therefore, it is practical to monitor the presence of forms or buttons on the webpage. Following the example depicted in Figure 12, we could search the visible text for the presence of words invoking the sense of urgency like “Urgent, Hurry Up, Don’t miss, Limited period,” etc. Other characteristics might be derived from the number of links (tag <HREF>) in the webpage – as phishing webpages try to reduce links pointing to external sites, as they want the victim to stay on site. Other characteristics might be derived from observing the presence of obfuscation or masking techniques of parts of the HTML page or linked scripts – this is often done to hide the actual functionality of scripts and make it hard to analyze.

While HTML 4.01 had approximately 90 tags, the current HTML 5 standard has approximately 140. In addition to the tags themselves, further characteristics can be derived from the combination or content of the text, placement, encoding, etc.

As part of our research, we consolidated a comprehensive list of 84 HTML-based features (see ***Appendix 2 – HTML-based Characteristics***).

2.2.4.3 Phishing Indicators – 3rd Party Data

Most common 3rd party data used for phishing are those linked to domain registrar – available via whois protocol. The domain's age is the most relevant and commonly used information for phishing. It is prevalent that phishing domains are fresh new domains that were registered only a short time before they were actually used. Therefore, the domain registration, expiration, and renewal dates are all very helpful in determining whether the page is phishing or legitimate. However, the practical complication lies in the fact that the date format depends on the registrar and registry [77]. Additional information, like registrant-related details (if present in the whois records), might also be used for phishing detection. A sample of whois response is available in ***Appendix 4 – Sample Raw Whois Response For uniza.sk***

In general, whois provides the following information:

- Registrant Information includes details about the individual or organization that has registered the domain. It typically contains the registrant's name, address, phone number, and email address. However, this information could be hidden or obfuscated.

- Registrar Information: Registrars are organizations accredited by the Internet Corporation for Assigned Names and Numbers (ICANN) or by a national ccTLD authority to register domain names.
- Domain Information: This includes important dates such as when the domain was registered and when it is due to expire. It also shows the domain's current status (e.g., active, reserved, or in dispute).
- DNS Information: Details about the domain's DNS settings, including nameservers, which help direct traffic intended for the domain to the correct server.
- Administrative and Technical Contacts: This section provides contact information for the people or organizations responsible for the domain's technical and administrative operations.

Another use-case for 3rd party data is using ping to identify the IP address of the domain. Ping is a diagnostic tool that tests the connectivity between two network nodes, such as a computer and a server. It sends Internet Control Message Protocol (ICMP) echo request packets to a specified address and waits for a reply. When the target device receives the echo request, it responds with an echo reply, allowing ping to measure the round-trip time it takes for the packet to travel to the destination and back. This measurement is reported in milliseconds and can indicate the network's performance or signal issues like packet loss.

```
root@LAMP ~# ping sme.sk -c3
PING sme.sk (104.22.13.230) 56(84) bytes of data.
64 bytes from 104.22.13.230 (104.22.13.230): icmp_seq=1 ttl=54 time=146 ms
64 bytes from 104.22.13.230 (104.22.13.230): icmp_seq=2 ttl=54 time=144 ms
64 bytes from 104.22.13.230 (104.22.13.230): icmp_seq=3 ttl=54 time=144 ms

--- sme.sk ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 144.027/144.621/145.652/0.731 ms
```

Figure 23 Sample ping command against sme.sk domain

The domain's IP address allows for the addition of additional third-party data to the mix. Via IP, we can add geo-location information about where the domain's hosting company is located and identify other domains hosted on the same IP. As part of our research, we consolidated a comprehensive list of 14 features based on the 3rd party data (see ***Appendix 3 – 3rd Party-based Characteristics***).

3 Design And Implementation Of Phishing Detection Solution

To achieve our research objectives, a high-level plan had to be prepared. This plan outlines the main phases of the overall project. These were further broken down into specific tasks ordered in a logical flow, with all their dependencies. Since a single person delivers the project, a detailed plan would not be required, as the work distribution aspect was unnecessary. Capturing the objectives and drafting the intended steps to achieve them helped keep the progress and pending activities apparent and transparent.

The planned phishing page detection solution is delivered right before the final project milestone, which is the completion of the thesis document. All the partial objectives fit in between and are linked with intermediary steps.

High-level steps and milestones:

1. **Phishing webpages source selection** – review and selection of possible sources of the data with fresh references to phishing webpages
2. **Phishing data capture** – capturing of the phishing webpages data, relevant fields extraction, and archiving
3. **Data transformation and cleansing** – data profiling, cleansing, standardization, enrichment, and preparation for the application of machine learning techniques
4. **Enrichment and features engineering** – extending the characteristics with additional data and deriving the features from the collected data (HTML, 3rd party data)
5. **Application of detection technique** – implementation of analytical models, review, and assessment of the KPIs
6. **Detection solution design and implementation** – selection of the most efficient technique and implementation into the detection solution
7. **Validation and experimentation with the detection solution**

3.1 Infrastructure And SW Technology Stack

All software technology is hosted on two physical servers:

- HP DL 380 G8 with 2 x Intel Xeon 24 cores @2.7GHz, 384 GB RAM, 2TB HDD, 2TB SSD
- DELL R730XD with 2 x Intel Xeon 16 cores @2.1GHz, 256 GB RAM, 1.5TB SSD

Both servers run the virtualization platform PROXMOX v8 in a 2-cluster node setup but without HA between the containers and virtual machines. The primary database is MariaDB v10.5, running as an LXC container. All web applications are hosted on a LAMP stack (Linux OS with Apache web server v2.4 and MariaDB as database server along with PHP v7.4 programming language) deployed as an LXC container. However, the Maria DB we used in the project was a standalone container, not the one deployed as part of the LAMP stack. Grafana v10.2 (PhishReport) is used for operational monitoring and is deployed as a standalone LXC container.

The main reason behind the selected software stack was previous experience with PHP and MySQL and their use for web scraping. Another aspect was easy migration and the small footprint of the stack.

3.2 Gathering The Data For Experiments

3.2.1 Sources Of Phishing Data

New phishing webpages are created and deployed daily, but there are only a few free and accessible data sources from which the data can be collected through automation. Though there are static datasets available for download [61]**Error! Reference source not found.**-[66], their usability is rather limited due to:

- **Freshness of the data -** many of the datasets were created several years ago. Phishing is continuously evolving; therefore, using stale data from a few years ago will certainly omit newer patterns and techniques or incorrectly represent the previous patterns that might have been suppressed in the current data. Such data might reduce the efficacy of models, misrepresent newer patterns, or even miss new types of phishing.
- **Transparency around the source, the process of collection, and applied data transformations** – this is a common pain point of many research papers, where the authors do not provide sufficient granular information related to how the original source data were collected, the origin of the data and also what data transformation or cleansing techniques were applied. Without the knowledge of these details, it is hard to evaluate the potential underlying issues within the data during the later stages

when these data are re-used for training the model. On top of that, it is also impossible to replicate or validate the results achieved.

- **The comprehensiveness of the data** - many datasets contain limited information. Most focus solely on the URL-derived characteristics, as those are the easiest to derive and work with. No datasets (except PhishMonger) include details about the web page characteristics.

Our objective was to assess the characteristics of phishing pages using as much information as it was practically possible to collect. We aimed to evaluate additional details from the phishing webpages, which required more comprehensive data collection. There are several free and publicly accessible databases continuously updated with newly reported phishing webpages. The most comprehensive data source is phishtank.org[1] , which was also selected as the primary data source for this thesis research.

3.2.1.1 PhishTank

In September 2006, PhishTank was started by the company OpenDNS as a free community website where people could post and verify phishing webpages. Later, in June 2015, Cisco acquired OpenDNS [13]. As it was and still is designed as a community project, all the phishing pages are reviewed by people (reviewers) who provide their assessment and make a final classification decision.

It is the most widely used source of phishing data (in [67], PhishTank was used in 25 out of 45 evaluated research papers. In contrast, the second most used data source was used in 6 papers, which shows how often PhishTank is being leveraged by the researchers). PhishTank provides data in a format in which the users report them; therefore, some reported URLs might contain typos or be formatted in an unusual way, which adds additional effort to the process of cleansing and standardizing the data. Registered users can participate in the manual review process of reported suspicious URLs and help classify them as confirmed phishing or legitimate webpage. Each submission has to be reviewed by more than one

[1] https://phishtank.org/

reviewer, and this number depends on the accuracy of the reviewer who flagged the submission as a phishing page. The reviewers with better accuracy will get better (higher) weights.

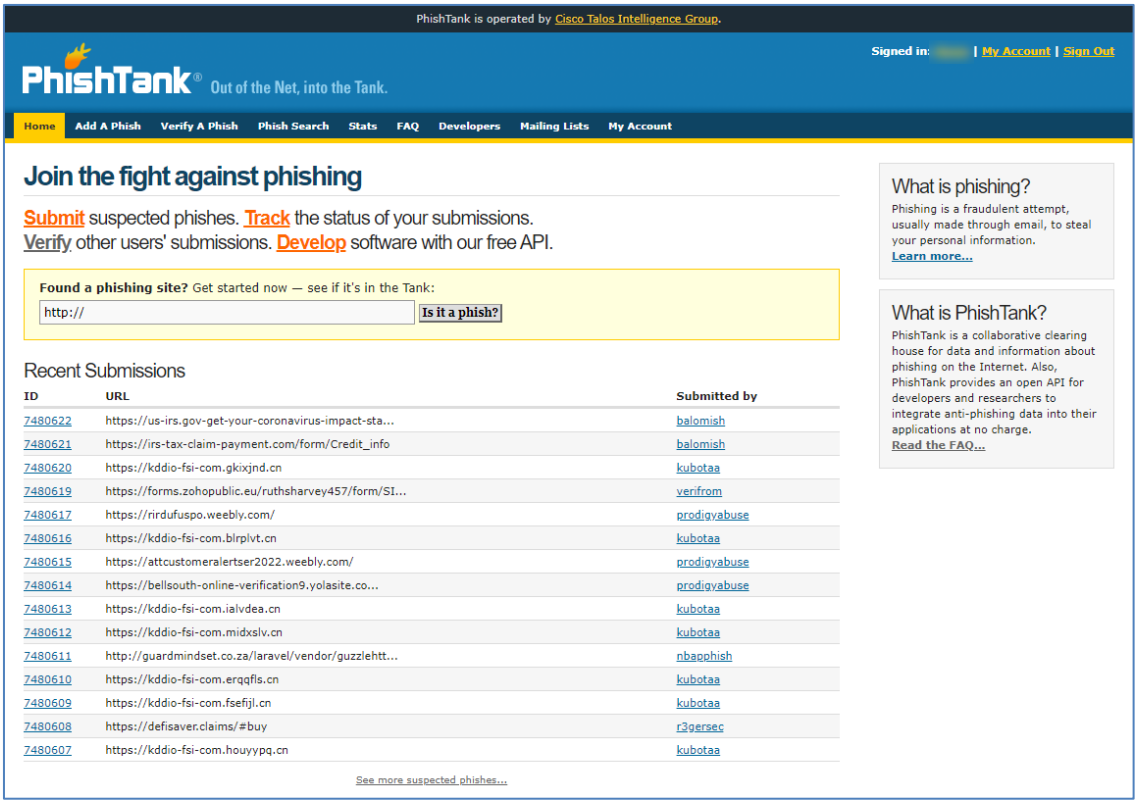


Figure 24 PhishTank website

The positive aspect of the manual classification approach is the highest possible classification accuracy. The negative side is a non-negligible volume of reported URLs that remain without the final classification (in Table 3, the daily volume of suspicious webpages in PhishTank is ≈700, but these are only records classified as confirmed phishing; the actual overall reported volume is ≈1150 records)

3.2.1.2 PhishStats

Started in 2014, though the archive data go back to 2009. PhishStats[2] receives the highest daily volume of reported phishing pages from all three selected data sources (Table

[2] https://phishstats.info

3). PhishStats also provides the most comprehensive number of characteristics for each reported URL, even though many are missing, and the actual details of how the characteristics are derived are not provided. PhishStats doesn't provide a webpage listing the data. These are available only through a comprehensive API.

3.2.1.3 OpenPhish[3]

Started in 2014 and is a free service providing a continuously updated feed of phishing URLs. Free service offers only basic information consisting of three columns - reported URL, targeted brand, and time when the URL was reported. There is an option to upgrade to a paid subscription, which provides more detailed information.

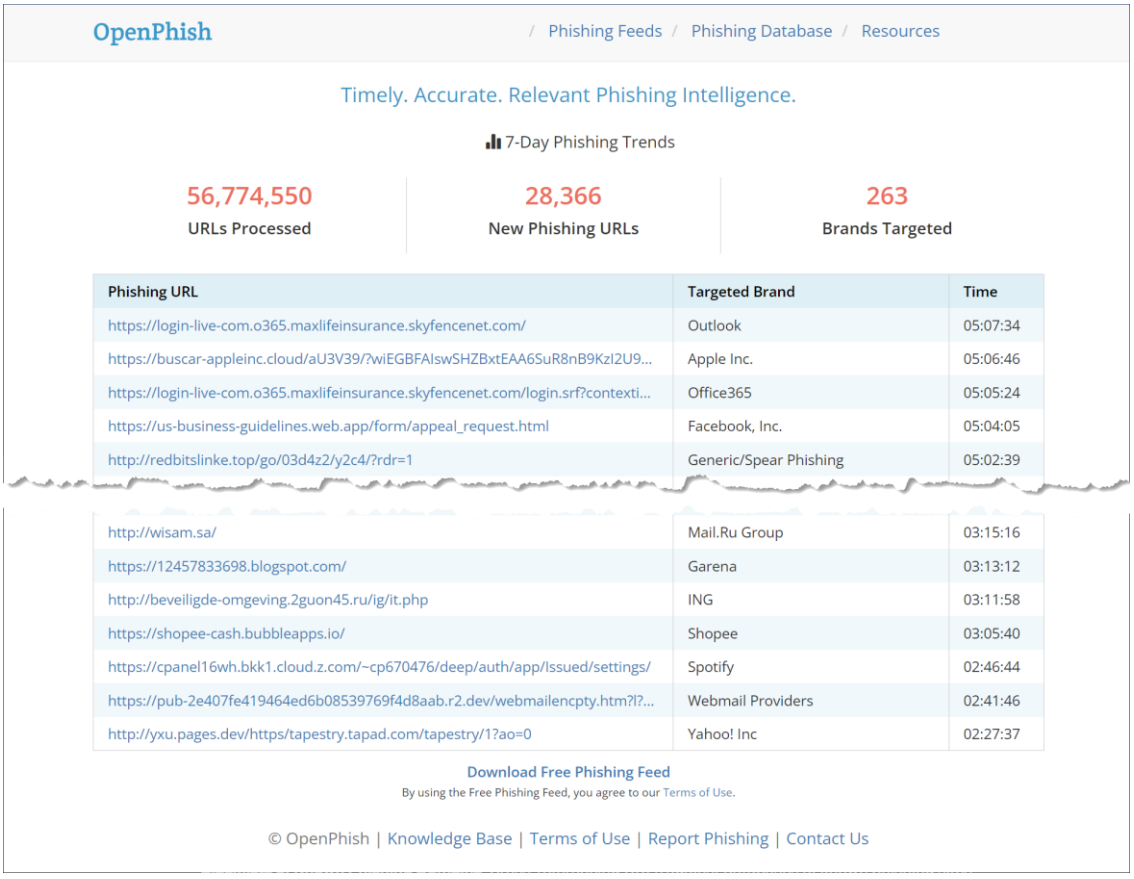


Figure 25 OpenPhish website

[3] https://openphish.com/

3.2.1.4 Datafeeds Comparison And Overlap Analysis

Phishing webpages can be reported via various channels, and the same suspicious URL can be shared or reported to various phishing lists, which causes data to overlap between these data sources. We analyzed the overlap between PhishTank, PhishStats, and OpenPhish, which can help decide on the preferred data source using these metrics. Table 3 briefly summarizes characteristics relevant to each of the discussed phishing data feeds.

Table 3 Comparison between PhishTank, PhishStats, and OpenPhish

	PhishTank	PhishStats	OpenPhish
Real-time interface	Web scrap	API	Web scraping
Batch interface	API	API	-
Records on website	Yes	No	Yes
Records Archive	accessible	accessible	inaccessible
Available features	***	*****	*
Daily volume[1]	≈700	≈2600	≈1000

[1] Daily volumes are calculated using the year 2023 data

As part of our research, we were able to collect PhishTank data as early as 2005, although we started the daily collection process in November 2021. For PhishStats, we obtained the entire archive from 2009, though we began the daily collection process in May 2022. For OpenPhish, we could not obtain an archive, so we have only data from the moment we started our collection process in April 2022 (all this data contains only URL-related information).

Analysis of overlap between PhishTank and PhishStats - for one of our conference papers [IS4], we analyzed data overlap between PhishTank and PhishStats. This analysis was conducted over a 10-year time period, from 2013 to 2022. The study considered only the confirmed phishing records. Each dataset was divided into separate monthly parts and de-duplicated, so each month-part contains the domain only once. The overlap was calculated by comparing all domain name lists (without the scheme and path parts as depicted in Figure 21) within these monthly parts of each data source.

The results (Figure 26) from the perspective of PhishTank data showed that almost all records from PhishTank data are present also in the PhishStats dataset with a visible drop (gap increased to approx. 15% from previous approx. 1%), which happened in 2017 and lasted till 2022 (while slowly closing down to approx. 4% in 2022). Further checking the recorded date

and time of the overlapped records showed that both datasets had the same date and time, meaning that PhishStats was possibly loading Phishtank data into its database.

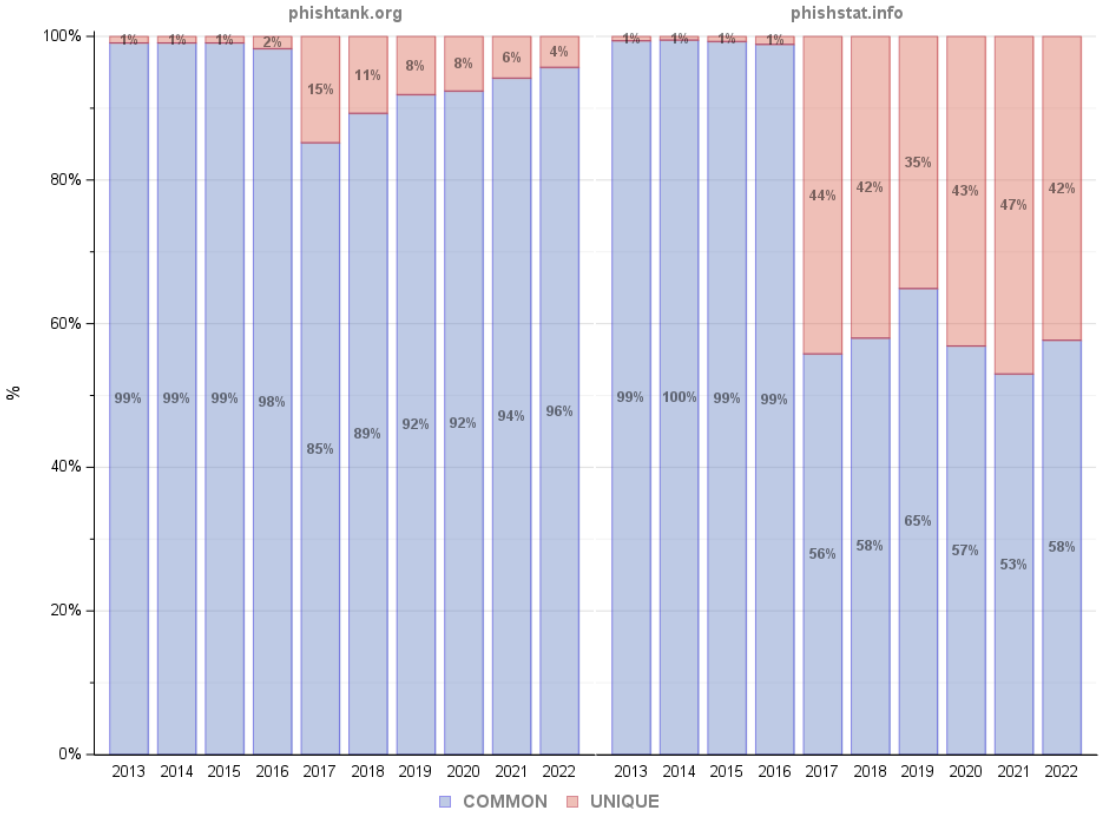


Figure 26 PhishTank and PhishStats data overlap for the period 2013 - 2022

From the PhishStats perspective, the data show that in the early years (2013 - 2016), PhishStats data were almost identical to Phistank's confirmed phishing data, and only starting from 2017 some additional sources were added. As the analysis was done with only confirmed phishing records in the PhishTank dataset, we performed an expanded analysis to verify whether these extra data are also not sourced from PhishTank (as the FP or UNK records). In this expanded analysis, all records for PhishTank and PhishStats were considered. Compared to the initial analysis, the volume of additional records in PhishStats was lower than the numbers (40%) shown in the initial analysis (see top right part of Figure 26). Still, the analysis confirmed that additional sources of phishing URLs were added during this period (2017-2022) - data not present in the PhishTank dataset.

This level of overlap - especially in the early years of 2013-2016 but also later - by merging the dataset would practically duplicate all PhishTank records and skew the results of any analysis if the data de-duplication would not be performed.

3.2.1.5 Analysis Of Overlap Between PhishTank, PhishStats, And OpenPhish

Since we didn't have a similarly long history for OpenPhish data and needed to understand the level of overlap with OpenPhish, we conducted a similar analysis with 2023 data only. We analyzed complete 2023 year data for all three data sources and followed the same approach described in the previous analysis between PhishTank and PhishStats. We divided one year of data into monthly parts and compared each month-part while using only the first five levels of the domain part of the URL. Match was found if all five domain levels (Figure 21) matched in the given month.

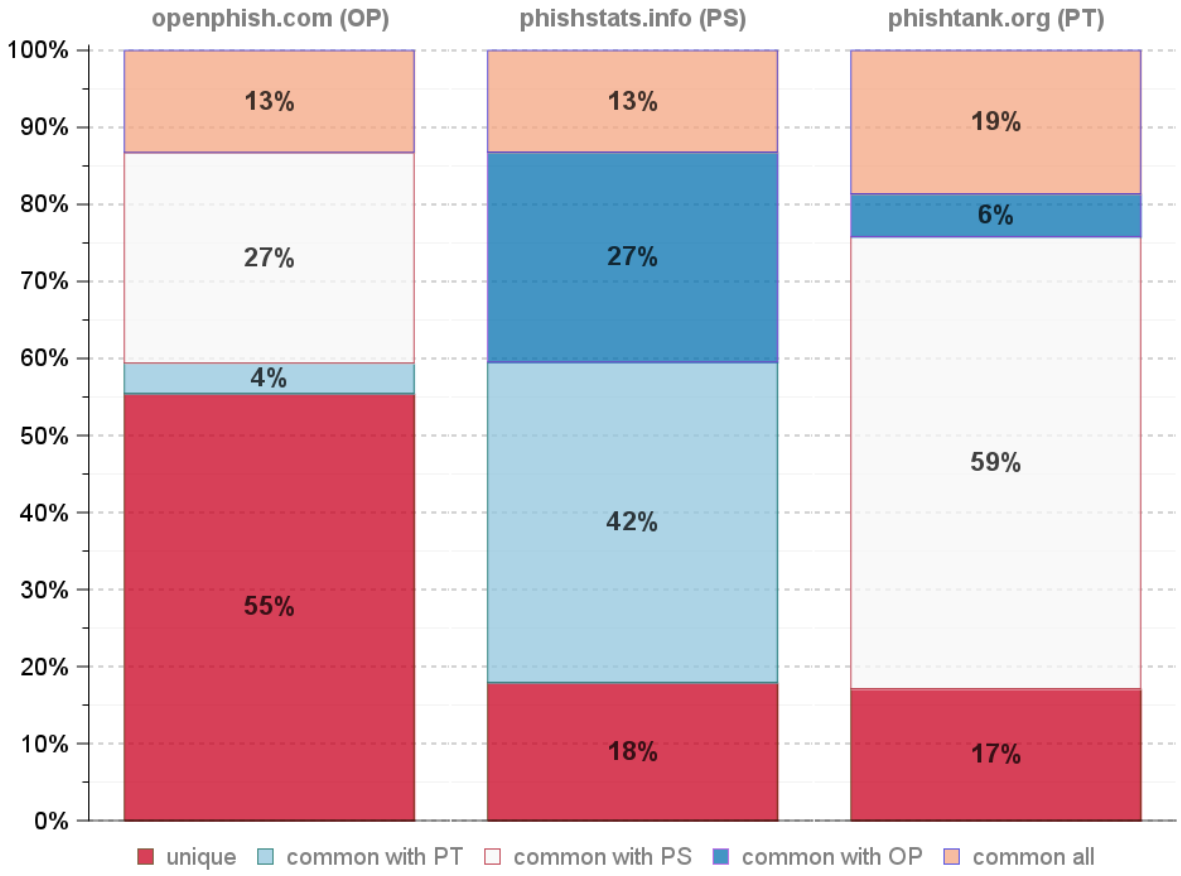


Figure 27 Data overlap between OpenPhish, PhishStats, and PhishTank on 2023 data

The results of the overlap analysis between the selected data sources show that the highest ratio of unique records has OpenPhish (Figure 27). And though the PhishStats has the highest daily volume, only 18% of records are unique (dark red color); the remaining 82% can also be found in PhishTank or OpenPhish.

3.2.2 Sources Of Legitimate Data

Phishing pages, though on the rise, constitute only a fraction of the 359 million domains across all top-level domains [68]. There are many ways to gather a sufficient volume of legitimate (non-phishing) webpages, but there are a few considerations to remember.

To train a predictive model, it is required to provide actual phishing data and equally relevant non-phishing data. In the research papers, we often see repeated instances of gathering the data from the following sources:

- **DMOZ** (dmoz.org) - also known as the Open Directory project owned by AOL and maintained by a community of volunteers (Figure 28). The web directory site used a hierarchical structure to organize site listings into categories and subcategories. AOL closed the project in 2017; since then, only archived old versions of the database have been available. DMOZ is often used as it contains URLs from diverse industries and countries, though the language prevalence is skewed with primarily English and European languages [69]. DMOZ was a relevant resource while it was maintained, though the URLs rarely contained the path and query part.
- **Alexa 1M** (alexa.com/topsites) - was a list of 1 million domains ranked by the traffic data collected via the Alexa toolbar and other traffic data sources. The list was often used as a reputation ranking database or Whitelist. The limitation of this list was that it contained only registered domain names (SLD.TLD components; Figure 21), which limited its suitability for deriving features based on URL characteristics directly from the list. The Alexa 1M list was discontinued in May 2022, but similar, there are several alternative lists - "Majestic Million" or "Umbrella 1 Million" from Cisco.
- **Yahoo** (yahoo.com) - another common source of URLs with legitimate webpages as it maintained its "Yahoo Directory" - a hierarchically organized database of links grouped into categories similar to DMOZ. Yahoo also provided another function that returned a

random URL from its directory. Both of the Yahoo functions were discontinued in December 2014.

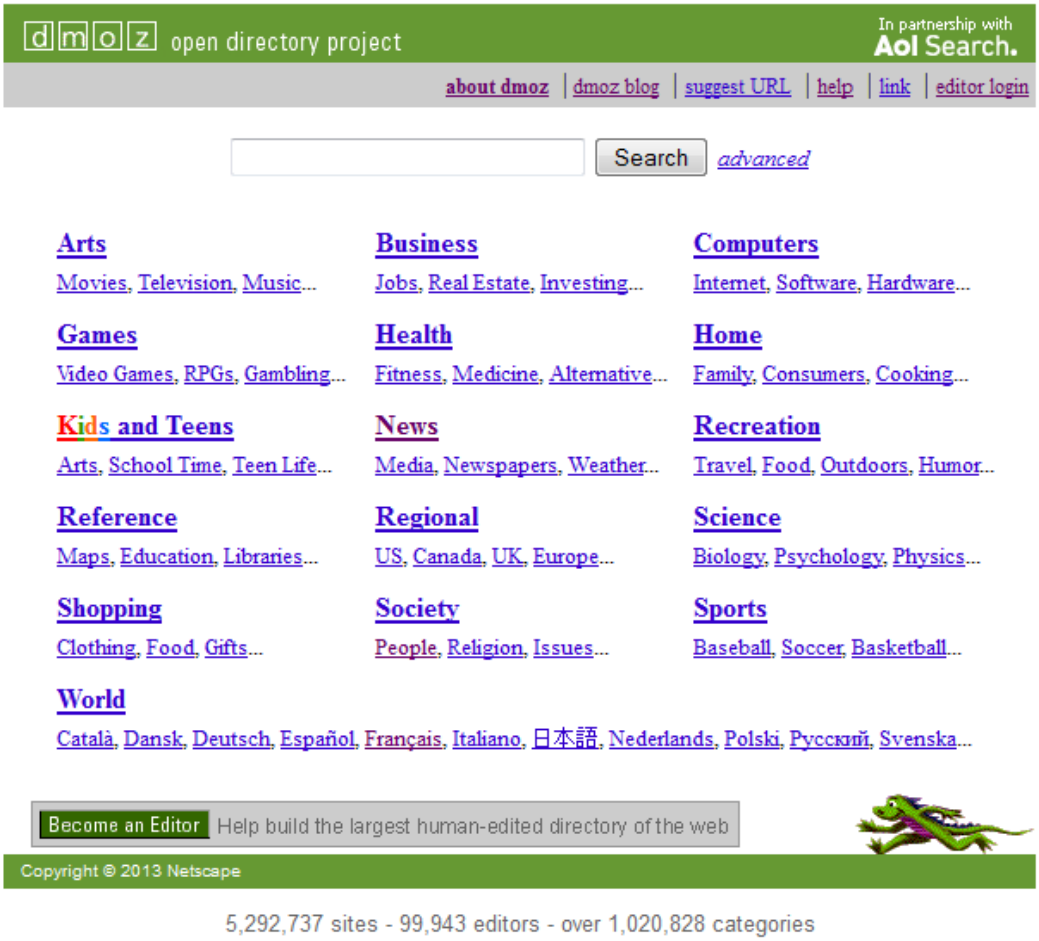


Figure 28 DMOZ Homepage in 2013; (dmoz.org)

- **Common Crawl** (commoncrawl.org) is a humongous web archive collected by automated crawlers containing billions of URLs spanning millions of domains. Common Crawl is a non-profit organization whose data are freely available and hosted on Amazon S3. They periodically crawl the web and create several snapshots(each containing more than 3 billion web pages) in one year. These data are structured and stored in three main formats:
    - the WARC (Web ARChive) file format, which contains the raw data, including HTML content, server response headers, and metadata;
    - the WAT (Web Archive Transformation) format, which provides metadata summaries of the contents of the WARC files;

- and the WET (Web archive Extraction of Text) format, which includes extracted plain text content from the web pages.

Part of the CommonCrawl's snapshot is an index. CommonCrawl index is a collection of 300 text-based files containing references/pointers within the actual WARC file containing the content (HTML) found when the spider was browsing and scraping the web. Each of these 300 text files requires between 5GB and 7GB of storage and contains between 12 million and 14 million URL records. The overall index requires ≈1.8TB of storage and holds ≈4 billion URLs.

Data from CommonCrawl are used for various purposes, such as training machine learning models (even Large Language Models like ChatGPT), analyzing web content, or studying internet trends. It is the most viable and comprehensive source of legitimate data, especially considering that the data reach back to 2008 and 2009.

3.2.3 Applications For Phishing Data Collection

Three web applications manage our collection of phishing data

- **PhishSearch** – responsible for accessing the source feeds, reading, parsing, and saving the newly reported URLs with all available details in the database in real-time
- **PhishCollect** – monitors the newly captured URLs from across all connected feeds and scrapes/gathers all relevant details for further processing. Collected details like a URL, landing page, scripts, favicons, etc., are stored as files.
- **PhishLongevity** – responsible for monitoring and capturing the status of the newly recorded URLs in the database in pre-defined time periods until the content hosted on the URL becomes unavailable. The webpage is then considered inactive.

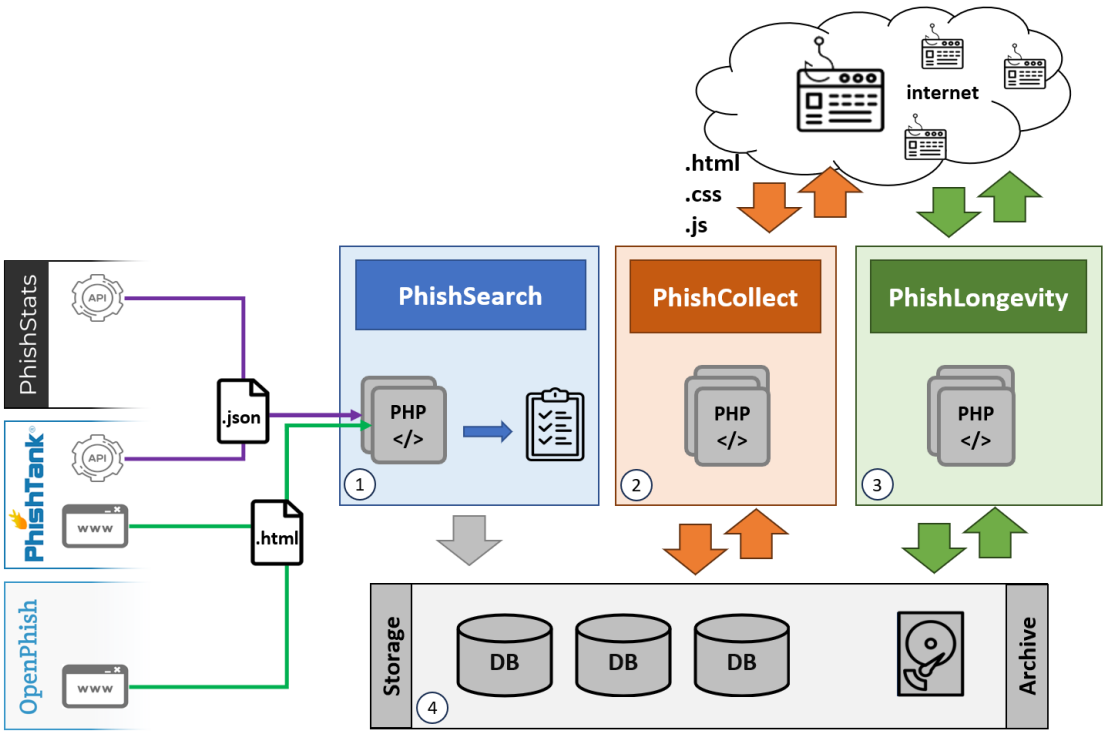


Figure 29 High-level logical architecture of data collection apps

3.2.3.1 PhishSearch

PhishStats provides an API interface that can be queried with various parameters to curate the returning set of data. Returned data in JSON format can be collected without any throttling, but a fair-use policy is requested to ensure availability for all interested users. The site also offers a full export of its database for 256 USD (with a 50% discount as per their web). PhishStats, in the current version, provides more than 40 descriptive attributes, which were gradually added throughout the years. Some are self-descriptive, but the logic behind populating others has to be guessed, as there is no documentation that would describe the process of populating all the provided columns.

PhishTank provides API interfaces, though it's more of a simple URL to download the confirmed phishing cases file. There are no options to manipulate or filter the returning list of records, as available with PhishStats. The returned data represent only a verified part of all recently reported URLs. Data received via API can be requested in JSON or XML format. There are also other data formats (CSV and serialized PHP).

A user must register on the PhishTank page to be allowed to pull the data from the API. The issue is that new registrations were not always available, as there were multiple downtime

windows between 2018 and 2023 when functionality was blocked. This was due to the long-standing plan to redesign the webpage entirely. The number of pull attempts is limited to less than ten pulls in one day. We scheduled the data-collection script to download the data every 6 hours (first, we used 8 hours frequency and later reduced it to 6 hours to get four data captures in a day).

PhishTank's JSON API interface returns a file with recent phishing webpages in a JSON format. Each dataset received from the JSON API contains confirmed phishing URLs from a range of dates. For example, the dataset requested on the 1st of December 2021 at around 6:00 am included 14075 records, but the dates ranged from 2008 to 2021. For details, see Figure 30.

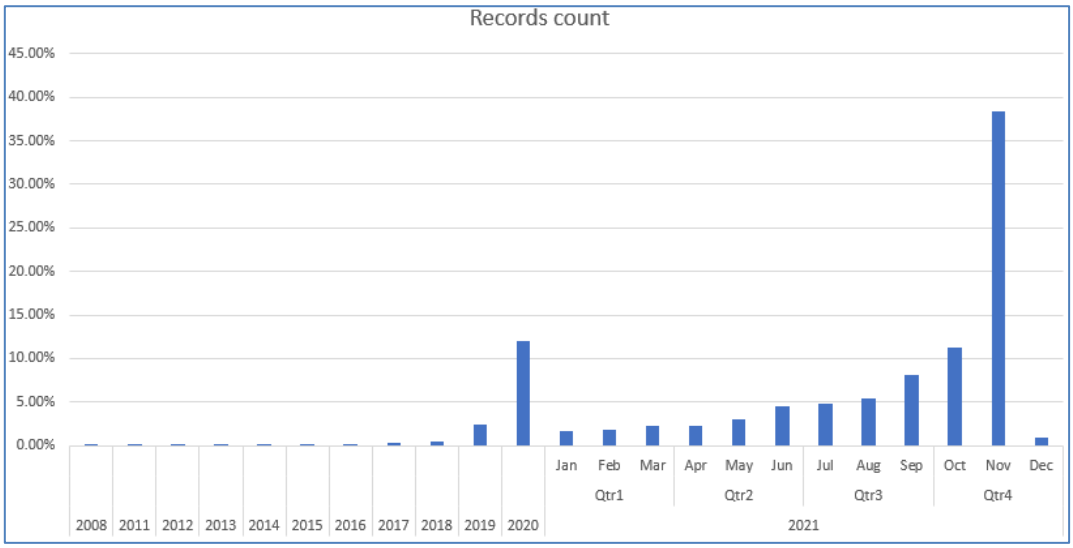


Figure 30 Distribution by date of submission in JSON provided data

One of the aspects related to API-provided data is the lack of a detailed explanation of the reason behind this wide range of data being provided. After reviewing the data from other dates, we could observe some patterns. The majority of the records are related to the most recent days, which could be logically explained by the fact that the submissions are being reviewed and marked as phishing or not continuously within a few days from the day of their submission. However, there is no clarity on why records from many years ago should still appear in the API data.

The following attributes are extracted from JSON and stored within the database table:

- **submission_time** - The date and time this URL was reported to PhishTank.

- **phish_id** - The ID number by which PhishTank refers to this submission.
- **url** – Reported suspicious URL.
- **phish_detail_url** – Link to PhishTank's detail summary page for this submission.
- **online** – Flag whether or not the phish is online and accessible. This is always ''yes'' in JSON data files since PhishTank only provides online phishes in these files.
- **verified** – Flag whether or not the community has confirmed this submission as phishing. This is always ''yes'' in JSON data files since PhishTank only supplies verified phishes in these files.
- **verification_time** – The date and time at which the submission was verified.
- **target** - The name of the company or brand the phish is impersonating if known.
- **details_ip_address** – IP address of the phishing webpage.
- **details_cidr_block** – CIDR IP address.
- **details_announcing_network** – Source from which the submission originates.
- **details_rir** – Regional internet registry authority.
- **details_country** – Country of IP location.
- **details_detail_time** – Date and time of the details data collection.
- **insert_record_dttm** – Date and time when the record was inserted into DB


```
[ ⊟
  { ⊟
    "phish_id":"7363094",
    "url":"https://dkddlur.weebly.com/",
    "phish_detail_url":"http://www.phishtank.com/phish_detail.php?phish_id=7363094",
    "submission_time":"2021-11-25T07:05:47+00:00",
    "verified":"yes",
    "verification_time":"2021-11-25T07:14:58+00:00",
    "online":"yes",
    "details":[ ⊟
      { ⊟
        "ip_address":"199.34.228.54",
        "cidr_block":"199.34.228.0\/22",
        "announcing_network":"27647",
        "rir":"arin",
        "country":"US",
        "detail_time":"2021-11-25T07:15:09+00:00"
      }
    ],
    "target":"Other"
  },
  { ...
  }
]
```


Figure 31 Sample structure of JSON response message with details of one record

OpenPhish was the last addition to the list of relevant data sources from which we collected the data. Data from OpenPhish are collected by periodically scraping the simple listing page with reported URLs.

PhishSearch is built using PHP programming language and operated as a web application through a web browser (Figure 33). The application would periodically (configured every 6 hours) pull data from the PhishTank's JSON API interface. Received data are then stored in two ways. The whole JSON dataset is stored as-is on the disk as a JSON file. Received data are parsed, and complete details are inserted into the dedicated MariaDB table.

Collecting data every 6 hours (Figure 32) poses a practical compromise. PhishTank limits the daily number of calls to 10, so 6 hours frequency resulted in 4 requests in 24 hours, leaving some buffer for ad-hoc requests. Reducing the frequency further was considered and declined as it was quite common for the request to fail. With a wider gap, we could potentially lose some records that would have already been removed from the confirmed phishing list JSON file with the next load.

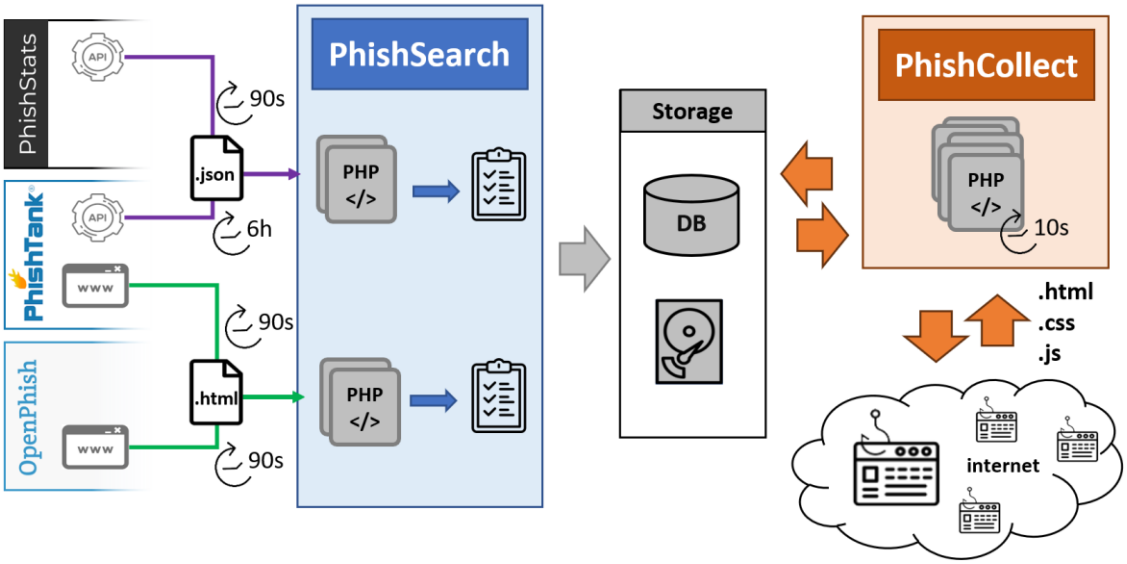


Figure 32 Data collection process via PhishSearch and PhishCollect

While gathering the phishing URLs from JSON, we observed that less than 50% of web pages were accessible for download. Therefore, we decided to collect the data more frequently and as soon as they become available in the PhishTank. As depicted in Figure 24, Phishtank provides a user interface via which it is possible to access and gather newly reported URLs. An HTML web scrapper was designed and implemented as part of PhishSearch to

address the limitations of the data capture process through PhishTank's JSON REST API. It was also confirmed that the phishing pages appeared in the PhishTank's API later, only after they were evaluated and flagged as confirmed phishing [9], but by that time, they were often already removed and inaccessible. Scraping the data from the web interface would remove this limitation (even though, at that time, we didn't know if the reported URL would be flagged as phishing). The functionality was added to the PhishSearch application, which allowed it to read and capture the details of the reported URLs every 90 seconds. The listing page contains the last 20 reported phishing URLs, and a 90-second pause between the data captures was established to mitigate the risk of the temporary IP block (see Chapter 3.2.3.3.2).

As part of our research, we presented at a conference [IS6] findings from our analysis (described in chapter *3.2.3.4 Capture Of Longevity – PhishLongevity),* which focused on phishing webpage longevity to better understand the time sensitivity and impact of time on the availability of the reported phishing webpages.

Figure 33 PhishSearch – depicting a capture of PhishTank JSON

After the data are captured from the data feed (PhishTank JSON+WEB, OpenPhish WEB, and PhishStats API), they are stored as files in the shared storage and later backed up into a monthly archive. After storing the data in physical files, details - like the URL of the phishing page - are extracted and saved into the database linked to PhishCollect.

This data collection process is automated, and the application for data capture is running continuously to capture the new phishing webpages submitted to PhishTank, PhishStats, and OpenPhish. However, occasional manual monitoring is needed to ensure that no part of the solution is failing or that no changes have been introduced into the source feeds - which does happen and usually causes interruption of the collection process.

On top of capturing the recently reported phishing pages, in the case of PhishTank, we also implemented the process to capture the historical records from previous years, though the usability of this data is limited. The reason is that though we could capture the data stored in PhishTank's archive (URL), the phishing webpages were mostly unavailable. Therefore, it would not be possible to calculate the descriptive characteristics derived from the webpage itself and its components. However, even partial data (URL) will allow us to perform an assessment of different techniques using URL data.

3.2.3.2 PhishCollect

PhishCollect is a PHP-based web application that periodically (every 10 seconds, but this is parametrized) monitors the status of the records in the database. If a new record is spotted, it captures a set of information. The process of reading the status and collecting details for any new records is depicted in Figure 32. Each instance of PhishCollect behaves as an independent agent that monitors specific tables for new URLs. When any new record is found, it will flag it (to prevent duplicate collection of this record by other agents) and proceed to collect the defined set of details. This process allows for parallel processing, so we can open many browser instances of the app and let them continuously monitor and capture the details of incoming URLs. In our production setup, we keep two virtual machines, each running ten instances of PhishCollect in parallel.

In the current configuration, PhishCollect captures the following:

- **Ping** – derives the IP address from the domain
- **IP Address geo-location** – collects the geo-location details from the IP address (continent, country, region, city, latitude, longitude, timezone, and zip code)
- **Screenshot** – captures the screenshot of the rendered webpage using googleapis.com
- **Favicon** – extracts any details and links to potential favicons on the webpage

- **Whois** – gathers details linked to the domain (registrar name, registrar IANA ID, Domain creation, update and expiration dates)
- **Landing page (HTML)** – saves the content of the landing page as an HTML file in original and UTF-8 encoding
- **HTTP header** – stores the HTTP header collected during the scrapping of the landing page (this header contains redirects)
- **Links** – extract links to all CSS, scripts, and HREF links

These details were collected to be used within our experiments when deriving and improving the detection model for this thesis and in the future.

3.2.3.3 Data Capture Problems

Though PhishTank provides JSON API and XML API for data collection, PhishStats also provides a JSON API interface for pulling the data; we faced several issues when collecting the data, which are summarized in the following sub-chapters.

3.2.3.3.1 Phishing Data Late Acquisition

Data available from JSON API contain only confirmed phishing pages. Evaluating whether the submitted webpage is a verified phishing page can last from a few seconds to hours, sometimes even a day or more. The life expectancy of phishing pages is decreasing rapidly. While some phishing attacks used to last days and weeks, today, this period is much shorter - 89% of domains linked to malicious activity had a lifespan of less than 24 hours and 94% less than three days [2]. Furthermore, when we attempted to download the phishing page after it appeared in the REST API call data, it was often too late, and the page was no longer available. This significantly reduced efficiency in gathering the webpage content data we planned to use for further analysis.

As the impact of this issue was rather strategic and would significantly hamper the ability to collect as much relevant phishing data as possible for further analysis, the solution was analyzed, found, and implemented. The actual solution was the new web scraping component. It captures the data directly from the webpage and ignores API data sources. This solution

resolved the problem of the outdated phishing URL being scrapped, but it unraveled another type of issue – query limit.

3.2.3.3.2 Web Scrapping Page Requests Limit

API interfaces are built for easy data provisioning. However, bypassing the API and scraping the phishing webpages directly from PhishTank's and OpenPhish's web interface uncovered a different issue.

Modern webpages are hosted with anti-DDOS solutions (often as part of CDN), which help prevent website unavailability due to DDOS attacks on domains (e.g., Cloudflare). The same solution also mitigates the abusive behavior of web crawlers or web scrappers flooding the web application server hosting the page with too many requests, resulting in the deterioration of webpage availability for all users.

This practically meant that if we tried to scrap many pages within a short period of time, the anti-DDOS solution temporarily blocked our access to the site. This behavior indicated that we had to implement a cooldown period (throttling) between web scrapping calls to ensure we didn't reach the request threshold and ended with a temporary block. On the other hand, this impacted the speed by which we could scrap the historical data, as a cooldown period had to be honored between every two requests.

3.2.3.3.3 Country-level Filtering

Data collection happens from Dubai in the United Arab Emirates (UAE). UAE has established a Telecommunications and Digital Government Authority (TDRA) that decides and maintains the content filtering lists at the country level. Internet providers have to implement these filters. The list of actual web pages that are being filtered (made unavailable from UAE) is not publicly disclosed. Still, the TDRA has released a report that indicates the volume of all web pages being filtered and categorized. From these statistics, we can observe that phishing category filtering increased between 2016 and 2018 (compared to other categories).

Table 4 Overall percentage of blocked websites as per TDRA[4]

PROHIBITED CONTENT CATEGORY	2016	2017	2018
Bypassing Blocked Content	0.18%	4.98%	5.34%
Pornography nudity and vice	71.90%	54.61%	44.19%
Impersonation fraud and phishing	11.23%	19.94%	30.35%
Insult slander and defamation	0.03%	0.10%	0.71%
Invasion of Privacy	0.05%	0.00%	0.30%
Supporting criminal acts and skills	0.00%	0.00%	0.00%
Drugs	1.65%	9.70%	4.06%
Medical and pharm. practices in violation of the laws	0.24%	0.12%	0.04%
Infringement of intellectual property rights	3.19%	3.38%	6.47%
Discrimination racism and contempt of religion	1.07%	0.65%	0.23%
Viruses and malicious programs	0.37%	0.59%	0.41%
Promotion of or trading in prohibited commodities	1.93%	0.63%	0.38%
Illegal communication services	3.71%	0.02%	0.08%
Gambling	2.27%	0.24%	0.04%
Terrorism	0.94%	1.17%	0.60%
Prohibited TLD	0.00%	0.00%	0.00%
Illegal Activities	0.05%	0.47%	2.63%
Upon order from judicial authorities	1.20%	3.40%	0.19%
Offences against the UAE and public order	0.00%	0.00%	3.99%
Total	**100%**	**100%**	**100%**

While scrapping the phishing data from the PhishTank website, we encountered that some of the submitted suspicious webpages are inaccessible from the UAE due to the above-described TDRA filtering.

Analysis performed on top of 2 months of recent data showed that, on average, 0.2% of submitted pages are impossible to scrap as the filtering blocks them on the country level. The solution to this problem would be to use a VPN or proxy server to perform the scrapping through a country that doesn't filter the internet traffic. Another solution would be to host

[4] https://tdra.gov.ae/-/media/Open-Data/Market-Information/English/Statistics-of-Prohibited-Content-Categories.ashx

the scrapping application on a server located in a country without content filtering (e.g., Slovakia).

3.2.3.3.4 Antivirus Filtering

Similar to the previous issue, though different from the source of the issue perspective, this issue is related to capturing phishing web pages and storing them in the computer's local storage.

The data capture process (also called web scrapping) reads the phishing page submission from PhishTank, PhishStats, or OpenPhish and reads the code of the webpage on that URL address. After reading, it stores the content into a file on a local disk. On top of the HTML page details, the algorithm also captures additional scripts, CSS, and references that it observes in the HTML page and saves them on a local disk as files. Some files were marked as suspicious or malicious as the data capture application used to run on a Windows laptop with active antivirus software (ESET Internet Security). As such, their saving was prohibited by the antivirus. This practically resulted in the inability to store the malicious content of the phishing webpages on a local disk as a file and use them later for analysis.

This problem was resolved by migrating the application to PROXMOX and LAMP stack on Debian, running without antivirus software.

3.2.3.4 Capture Of Longevity – PhishLongevity

PhishLongevity was implemented to answer a curiosity question and quantify the actual longevity and impact of delayed data capture on the ability to gather relevant data for a meaningful portion of reported URLs. The availability of quality and accurate phishing data is a long-lasting problem despite the existence of several websites from which the data can be comfortably captured. One of the reasons behind the limited availability of datasets with phishing data (beyond just the reported URLs) is also the short phishing webpage longevity or phishing webpage lifespan. There wasn't any dedicated research on this topic, though we found several statements capturing phishing longevity within published research articles.

PhishLongevity is a PHP + Maria DB-based web application that allowed us to collect granular phishing page data and analyze time sensitivity and the impact of time on the availability of the reported phishing webpages. It is also the 3rd data collection app after

PhishSearch and PhishCollect. PhishLongevity (in Figure 34, represented by a blue square named PhishLongevity) contains two sub-components – **Identify** and **Verify. Identify** works in a similar fashion as PhishSearch and Verify mimics PhishCollect. The overall process can be summarized as:

1. Attacker deploys new phishing webpage (T_0)
2. A user observes a suspicious phishing webpage
3. The user reports the suspicious webpage to OpenPhish or PhishStats (T_1)
4. **Identify** checks the phishing feed and records the new webpages (T_2)

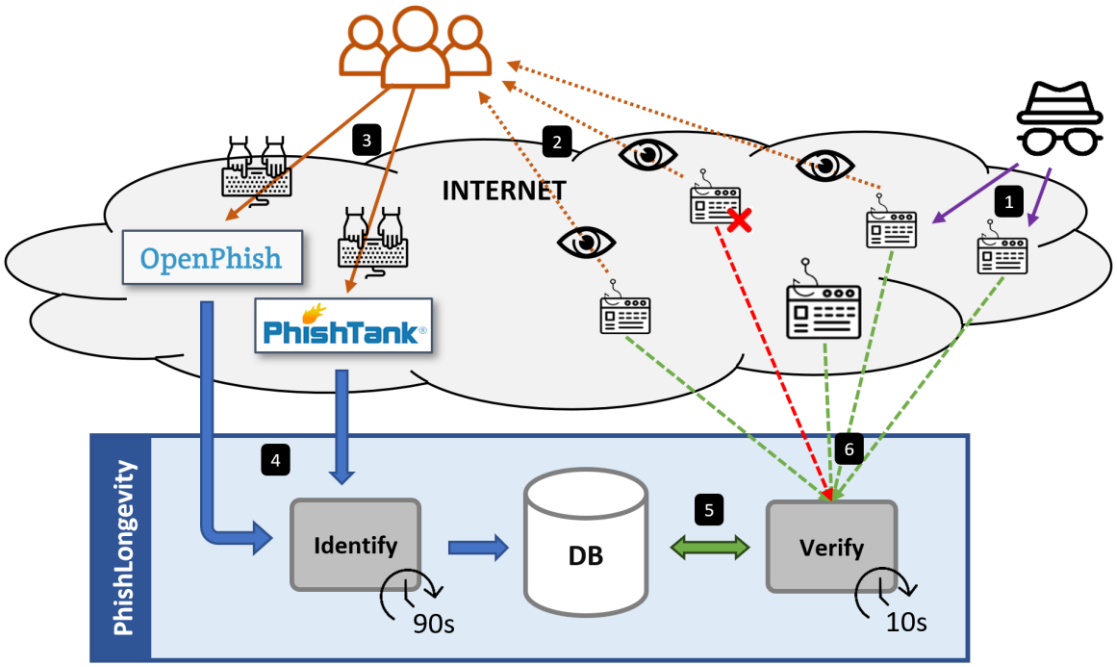


Figure 34 PhishLongevity - data collection process diagram

5. **Verify** reads the newly recorded suspicious URL in the database, navigates to the URL, and captures its status - active or inactive (T_3)
6. **Verify** continues to read the suspicious webpage in a pre-determined time window until the page is observed as inactive.

Identify - was responsible for reading the data feeds every 90 seconds and recording new suspicious phishing URLs. This threshold was decided to balance the requirement to capture the status of phishing webpages as soon as possible with the impact of frequent reading on the underlying infrastructure. New records were saved into the database table,

which utilized the uni-temporal structure of the destination table to capture the details while separating inactive pages into the "HIST" table.

Verify - was periodically (every 10 seconds) monitoring the status of the records in the database and capturing the status if the record was new or the pre-defined time since the last status check had passed, and during the previous check, the webpage was still accessible (active). Verify stopped checking the status of those records that were inactive during the last status check (in practical terms, when the document size - HTML - read from the URL was zero, which meant it was removed or the hosting provider blocked the URL/site).

In the deployed configuration, we defined the following time windows at which the status was captured:

- immediately as the URL was recorded
- every hour within the first 24 hours
- every day within the first 14 days
- 3rd and 4th week of the first month
- 2nd - 6th, 9th and 12th month

3.2.3.5 Data Collection Process Monitoring – PhishReport

Despite automating the data collection process, it is not uncommon to observe an interruption for various reasons. The interruptions often result from service unavailability or changes done on the data provider side (PhishTank, PhishStats, OpenPhish). Sometimes, the interruptions happened on our side when browsers auto-updated and didn't continue where they were interrupted. Or containers hosting web browsers have crashed or other – primarily technical – reasons.

To mitigate the impact of such interruptions to a minimum and observe them as soon as they happen, we implemented a single-screen overview dashboard that quickly and visually shows any disruption in the collection process. This dashboard was built using Grafana – an open-source analytics and interactive visualization web application that provides charts, graphs, and alerts for the web when connected to supported data sources. It is primarily used for monitoring and observing time-series data. Grafana allowed us to create a dashboard with

a wide variety of visualization options, as seen in Figure 35. Grafana was deployed into Proxmox as an lxc container

Figure 35 PhishSearch and PhishCollect operations monitoring dashboard

The first row shows the status of PhishCollect via three graphs (left to right):

- The volume of records processed in the last three days (captured in intra-day and historical table)
- Current volumes of records in intra-day table by status
- The volume of scrapped URLs for the last 24 hours by an hour

The next three rows visualize hourly, daily, and monthly volumes of records from PhishSearch for PhishTank, OpenPhish, and PhistStats (from left to right).

3.2.4 Applications For Legitimate Data Collection

In the same way, we collect various characteristics for phishing data – as described in the above chapters - we implemented a separate web application called URLCollect to collect similar characteristics for legitimate pages. From a design perspective, URLCollect provides analogous functionality to PhishCollect, but there is a difference in the process of reading the source data. PhishCollect was constantly monitoring a specific database table to spot newly inserted URLs. This behavior was required to collect the details of phishing web pages as soon

as possible due to the limited lifespan of the web pages. Legitimate pages don't need such an approach.

The most relevant legitimate data sources are Common Crawl or alternatives to Alexa 1M, like Cisco Umbrella 1M. However, these data are provided as a list of URLs; therefore, we designed the URLCollect to identify specific database tables in the database schema (via a particular prefix to the table's name). These tables are manually prepared from whichever source we consider relevant, and each has to have two mandatory columns – URL column and ID column. URLCollect will then start collecting the data on the provided URL located in the identified URL Column from the selected table, one observation after another, using the ID column.

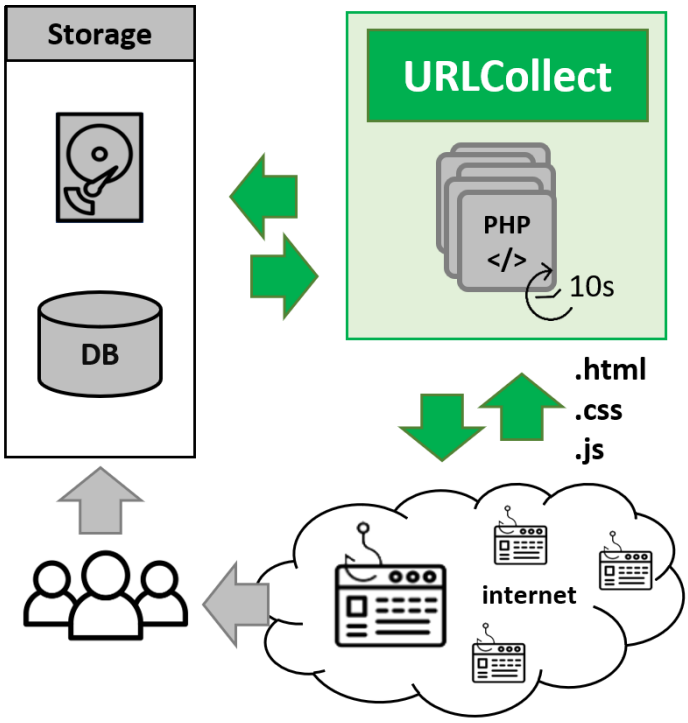


Figure 36 Data collection process via URLCollect

From an operational perspective, we usually run multiple URLCollect instances, each in its web browser tab, and we collect the details in parallel to speed up the overall process. There was no need to deploy any throttling as each record points to a different domain.

3.3 Training And Testing Dataset Preparation

Despite a lot of research in this area, it is challenging, if not impossible, to compare the results of one study with another. There are various reasons, but the main ones observed are the insufficient level of detail about the source of data (many times, the details about gathering the legitimate records are inadequate or missing) used by these studies or the lack of details related to data transformation and cleansing before using machine learning techniques [78]. Studies often overlook the importance of describing the data collection process and the adjustments performed, which are crucial to validate or compare the results between various researchers.

To allow for easy comparability of various kinds of research, we put together a comprehensive framework for preparing a dataset that could be used to train a predictive model for phishing detection [IS7]. In the research, we described the whole process of preparing a balanced and comprehensive dataset to provide the best possible outcomes, along with guidelines and considerations that should be included in the design process.

3.3.1 A Framework For Preparing A Balanced And Comprehensive Dataset

Creating a viable dataset for training predictive models starts with selecting data sources. We identified the three free data feeds described in Chapter *3.2.1* – PhishTank, OpenPhish, and PhishStats. In the framework, we also listed various options for legitimate pages, which are also described in Chapter *3.2.2.* Then, we summarized the usual data cleansing techniques (e.g., de-duplication) to ensure the data quality and accuracy in the dataset. The next part described the process and areas of focus for the creation of derived features – the same as the ones described in sub-chapters of Chapter *2.2.4.* The next step was choosing the optimal size of the dataset and identified impacting factors were:

a) training and testing set of data - we require a sufficient number of observations to ensure we can train the model on one set of data and validate it on another set
b) type of algorithm used - some algorithms, like neural networks, can efficiently ingest and also usually use larger datasets for effective training compared to, e.g., decision trees, which can partition the space and train the model on smaller datasets.

c) data diversity – to ensure the proper generalization of the model, it is crucial to warrant the sufficient diversity of phishing attacks in the underlying data. Sufficient diversity also means that the prevalence of the derived indicators from these various phishing attacks would ideally represent the prevalence of those attacks in the real world.

d) dimensionality of the data – the number of features/characteristics we are planning to use to train the model will impact the needed size of the data. More features often require more data samples to accurately model the prevalence of values for all the features and their relationships.

e) practical data availability – this is highly relevant to particular sub-classes of phishing (like spear phishing or phishing against specific uncommon types of industries, or when we plan to do comparative analysis further back to the past, etc.) where the availability of legitimate and phishing examples also constrains size. Real-world data availability might limit the dataset size.

Among the researchers are those who use a few hundred records for each class [79], those who use a bit more than a thousand records [73], those who use a few thousand [80], and then a few who use tens of thousands of records [61]. Using a few hundred or thousand records might not be sufficient, especially considering the aspects mentioned above. It is possible to conduct a simple exercise that starts training the model with the smaller size of the data and gradually increases and observes the change in the KPIs (True-Positive Ratio, False-Positive Ratio, Accuracy, Balanced accuracy, F-1 score, etc.). We should observe decreasing gains as the data volume is increased to the point where no further data increase will positively impact the results. The bigger the dataset, the better the detection outcome, as stated in [78], is not necessarily always true. The more representative the dataset, the more comprehensive the features collected and the better the detection performance [81].

We experimented by evaluating balanced accuracy as our primary performance indicator across various training dataset sizes and feature set sizes by training three models – Logistic regression, Decision Tree, and Support Vector Machines. Analysis was conducted using an existing dataset [84] with 58,000 records of legitimate webpages and 30,647 phishing

webpages. The dataset contains a column indicating whether the record is phishing or a legitimate webpage and another 111 derived features.

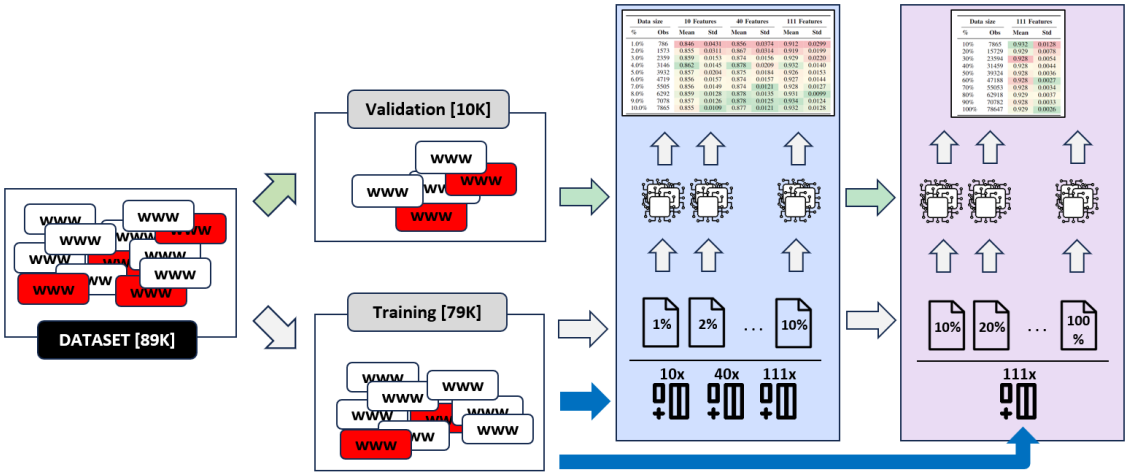


Figure 37 Analysis of dataset size and dimensionality impact on model accuracy

We separated a validation dataset of 10,000 records from the original dataset holding ≈89K records while keeping the phishing and legitimate records ratio as it exists in the dataset (1:1.9). The remaining data (≈79K) were used to train models using various training dataset sizes. In the first step, we evaluated model accuracy by using a training dataset of **1% to 10%** of the size of the dataset (blue-colored rectangle in Figure 37). At the same time, we evaluated the model accuracy for the dimensionality of the data (10x, 40x, and 111x representing the number of features used in the training). In the second step, we trained the models using **10% to 100%** of the dataset size to see whether further increasing the number of records contributes to even better models (purple-colored rectangle in Figure 37 while using only 111 features).

The results of the combination of various sizes and features for logistic regression are in Table 5, for decision tree in Table 6, and for SVM in Table 7. In the results, we observed the positive impact of the size, especially within the size between 1% and 4% of the dataset size. Gradual improvements across all three models, as well as across all feature variants, can be observed. In the 5% and 10% range, we observed mixed results, where only the decision tree algorithm gradually improved. At the same time, the remaining two models slightly deteriorate, though we observe the improvement of standard deviation figures.

Table 5 Performance across training data sizes and feature counts – log. regression

Data size		10 features		40 features		111 features		Data size		111 columns	
%	Obs	Mean	Std	Mean	Std	Mean	Std	%	Obs	Mean	Std
1%	786	0.846	0.043	0.856	0.037	0.912	0.030	10%	8k	0.932	0.013
2%	1573	0.855	0.031	0.867	0.031	0.919	0.020	20%	15k	0.929	0.008
3%	2359	0.859	0.015	0.874	0.016	0.929	0.022	30%	23k	0.928	0.005
4%	3146	0.862	0.015	0.878	0.021	0.932	0.014	40%	31k	0.928	0.004
5%	3932	0.857	0.020	0.875	0.018	0.926	0.015	50%	38k	0.928	0.004
6%	4719	0.856	0.016	0.874	0.016	0.927	0.014	60%	47k	0.928	0.003
7%	5505	0.856	0.015	0.874	0.012	0.928	0.013	70%	55k	0.928	0.003
8%	6292	0.859	0.013	0.878	0.014	0.931	0.010	80%	63k	0.929	0.004
9%	7078	0.857	0.013	0.878	0.013	0.934	0.012	90%	71k	0.928	0.003
10%	7865	0.855	0.011	0.877	0.012	0.932	0.013	100%	79k	0.929	0.003

Table 6 Performance across training data sizes and feature counts – decision tree.

Data size		10 features		40 features		111 features		Data size		111 columns	
%	Obs	Mean	Std	Mean	Std	Mean	Std	%	Obs	Mean	Std
1%	786	0.861	0.034	0.849	0.034	0.910	0.031	10%	8k	0.929	0.013
2%	1573	0.872	0.028	0.873	0.019	0.909	0.020	20%	15k	0.932	0.005
3%	2359	0.881	0.020	0.877	0.018	0.922	0.017	30%	23k	0.935	0.006
4%	3146	0.881	0.018	0.887	0.016	0.929	0.016	40%	31k	0.941	0.005
5%	3932	0.881	0.025	0.879	0.022	0.921	0.019	50%	38k	0.943	0.003
6%	4719	0.876	0.020	0.887	0.021	0.922	0.012	60%	47k	0.944	0.002
7%	5505	0.880	0.019	0.881	0.022	0.924	0.015	70%	55k	0.946	0.003
8%	6292	0.885	0.015	0.885	0.012	0.924	0.010	80%	63k	0.947	0.003
9%	7078	0.890	0.011	0.887	0.015	0.929	0.010	90%	71k	0.947	0.003
10%	7865	0.886	0.009	0.891	0.012	0.929	0.013	100%	79k	0.950	0.003

Table 7 Performance across training data sizes and feature counts – SVM.

Data size		10 features		40 features		111 features		Data size		111 columns	
%	Obs	Mean	Std	Mean	Std	Mean	Std	%	Obs	Mean	Std
1%	786	0.836	0.049	0.861	0.039	0.901	0.038	10%	8k	0.932	0.012
2%	1573	0.854	0.031	0.864	0.031	0.919	0.021	20%	15k	0.930	0.008
3%	2359	0.858	0.015	0.871	0.014	0.929	0.022	30%	23k	0.929	0.005
4%	3146	0.864	0.014	0.877	0.020	0.929	0.013	40%	31k	0.929	0.005
5%	3932	0.859	0.021	0.872	0.021	0.926	0.014	50%	38k	0.929	0.004
6%	4719	0.857	0.016	0.870	0.018	0.925	0.013	60%	47k	0.928	0.003
7%	5505	0.856	0.016	0.871	0.014	0.930	0.014	70%	55k	0.929	0.003
8%	6292	0.857	0.014	0.874	0.011	0.931	0.011	80%	63k	0.929	0.004
9%	7078	0.857	0.014	0.874	0.013	0.934	0.011	90%	71k	0.929	0.003
10%	7865	0.854	0.012	0.872	0.015	0.932	0.012	100%	79k	0.929	0.003

The analysis of the impact of dimensionality is relatively straightforward. We see that an increased number of features brought incremental gain in accuracy and reduced standard deviation. However, the added features must be relevant and have unique characteristics complementing the other features. It is also important to note that some algorithms are more sensitive to higher dimensionality (e.g., Support Vector Machine compared to the other two algorithms) and might result in increased training time needed, even to the point that would not be practical.

Via the experiments, we also confirmed that more features might require a bigger dataset, which is visible when we compare the best result achieved with the dataset with only 10 features with the best result achieved for the dataset having all 111 features. While the dataset with the smallest number of features achieved its best result with the dataset of 4% size, the entire dataset with 111 features achieved the best results with the 9%-10% sized dataset. This also confirms a logical assumption that a dataset with more features would require more data observations to provide samples for all relevant combinations of these features.

The framework's last step focuses on the dataset's internal structure. The previous section stated that having more patterns available within the training data allows the trained model to approximate the underlying correlations better and, therefore, be more accurate when classifying new records. The structure of the data also impacts the variability of the patterns. The structure of the data means understanding the share of industries targeted by phishing, as some are more prevalent than others. It also means looking at the language of the phishing targets. But, the first structural decision concerns the dataset's ratio between phishing and legitimate records. We would get a hugely imbalanced dataset if we collected all the URLs on the web and could identify all the phishing pages among these URLs. The ratio between legitimate and phishing web pages could easily be 1:1000 or even more. Therefore, what should the ratio between phishing and non-phishing pages in the dataset be? Researchers have asked the same question in [82], and they decided to train the data on a balanced dataset, but evaluation and testing were performed on an imbalanced dataset. In general, it is advised to construct and train the model on a balanced dataset so that the algorithm can have an equal chance to extract the characteristics of phishing pages and those legitimate. The balanced dataset was also used in [61] and [80]. In [83], researchers performed

an analysis where they calculated the True-Positive Rate (TPR) and False-Positive Rate (FPR) for various ratios of phishing records in the dataset. The result of this analysis was that the TPR grew gradually from 93% to 98% for 10% to 50% and stayed almost the same for the 60% and 70% ratio of phishing records in the dataset, but at the same time, the FPR grew from 0.5% to 1.25% from 10% share to 50% share and continued to grow to 2% for 70%. Researchers in [79] performed a test with two different ratios of legitimate vs. phishing - 60:40 and 82:18. The outcome was that the PhiDMA algorithm performed with higher accuracy on more skewed data. But, since Accuracy as a qualitative measure doesn't perform well with skewed data, we also calculated balanced accuracy, which also performed slightly better for a more skewed ratio of 82:18 (95.63% vs. 92.36%).

To evaluate the impact of the ratio on the model's accuracy, we conducted another experiment where we trained three models – logistic regression, decision tree, and SVM using varying shares of phishing and legitimate records in the dataset while using only 10 first features from the dataset. Analysis was conducted using the dataset described in the previous section [84]. We separated 10,000 records from the dataset used as a validation dataset. We created a balanced dataset from the remaining data containing 30,000 legitimate and 30,000 phishing records. This dataset of 60,000 records was used as a pool from which we derived the training dataset used to train the models. All three models were trained on top of the freshly created dataset with 30,000 records while varying the ratios of legitimate and phishing records - starting with 90% legitimate and 10% phishing and gradually moving towards 10% legitimate and 90% phishing. We used the smallest number of features - the first 10 - and gathered the model's mean balanced accuracy figures - similar to the previous analysis.

The results of the experiments are available in Table 8 for the Logistic regression model, Table 9 for the Decision Tree model, and Table 10 for the Support Vector Machine model. In the results, we observed the best results around the balanced ratio only for the Decision Tree model. In the results, we can also observe that the number of phishing records in the dataset results in very similar balanced accuracy figures across various sizes of datasets and ratios of phishing records. For regression and SVM, the results show that a higher ratio of phishing records positively impacts the balanced accuracy of the trained model.

Table 8 Performance across training data sizes and ratios – log. regression.

Training data		Ratio of phishing and legitimate records within training dataset								
%	Obs	10:90	20:80	30:70	40:60	50:50	60:40	70:30	80:20	90:10
10%	7865	0.767	0.781	0.836	0.847	0.857	0.888	0.891	0.895	0.896
20%	15729	0.750	0.784	0.842	0.848	0.855	0.876	0.890	0.890	0.881
30%	23594	0.755	0.788	0.842	0.852	0.855	0.879	0.883	0.887	0.893
40%	31459	0.761	0.794	0.841	0.852	0.855	0.877	0.890	0.887	0.885
50%	39324	0.754	0.792	0.841	0.855	0.852	0.875	0.891	0.889	0.892
60%	47188	0.749	0.789	0.840	0.852	0.854	0.881	0.891	0.887	0.895
70%	55053	0.751	0.792	0.844	0.850	0.856	0.878	0.888	0.892	0.888
80%	62918	0.746	0.789	0.840	0.850	0.856	0.877	0.886	0.888	0.893
90%	70782	0.754	0.787	0.844	0.852	0.855	0.878	0.888	0.888	0.889
100%	78647	0.755	0.788	0.841	0.852	0.857	0.878	0.888	0.888	0.891

Table 9 Performance across training data sizes and ratios – decision tree.

Training data		Ratio of phishing and legitimate records within training dataset								
%	Obs	10:90	20:80	30:70	40:60	50:50	60:40	70:30	80:20	90:10
10%	7865	0.749	0.833	0.874	0.888	0.893	0.888	0.886	0.892	0.894
20%	15729	0.765	0.824	0.881	0.884	0.885	0.892	0.890	0.882	0.886
30%	23594	0.752	0.837	0.865	0.888	0.897	0.892	0.893	0.890	0.894
40%	31459	0.766	0.838	0.869	0.891	0.894	0.891	0.893	0.886	0.886
50%	39324	0.765	0.836	0.872	0.893	0.895	0.894	0.895	0.888	0.890
60%	47188	0.775	0.834	0.871	0.895	0.894	0.895	0.892	0.896	0.894
70%	55053	0.779	0.835	0.888	0.892	0.895	0.896	0.893	0.893	0.888
80%	62918	0.776	0.837	0.873	0.895	0.896	0.896	0.896	0.889	0.887
90%	70782	0.779	0.837	0.887	0.897	0.897	0.896	0.896	0.893	0.889
100%	78647	0.784	0.833	0.873	0.895	0.896	0.895	0.892	0.893	0.891

Table 10 Performance across training data sizes and ratios – SVM.

Training data		Ratio of phishing and legitimate records within training dataset								
%	Obs	10:90	20:80	30:70	40:60	50:50	60:40	70:30	80:20	90:10
10%	7865	0.695	0.790	0.838	0.853	0.860	0.855	0.879	0.891	0.874
20%	15729	0.696	0.789	0.834	0.849	0.858	0.858	0.891	0.891	0.892
30%	23594	0.698	0.794	0.842	0.856	0.854	0.855	0.887	0.889	0.889
40%	31459	0.708	0.795	0.843	0.852	0.854	0.854	0.883	0.887	0.891
50%	39324	0.708	0.789	0.845	0.857	0.851	0.857	0.882	0.892	0.890
60%	47188	0.699	0.795	0.842	0.850	0.851	0.859	0.888	0.891	0.899
70%	55053	0.715	0.789	0.843	0.854	0.854	0.856	0.884	0.891	0.888
80%	62918	0.692	0.793	0.841	0.852	0.852	0.857	0.885	0.890	0.886
90%	70782	0.697	0.792	0.841	0.855	0.852	0.856	0.885	0.889	0.890
100%	78647	0.700	0.793	0.842	0.849	0.854	0.855	0.882	0.889	0.889

While for 10 features, we observed in the first analysis that the model didn't improve further beyond the 3000 records dataset (this dataset had a ratio of phishing vs. legitimate records 1:1.9) and balanced accuracy 0.862, in the second experiment with varying ratios, we achieved even higher balanced accuracy as we moved to the higher ratio of phishing records within the dataset across all dataset sizes. The same results were achieved for SVM. Training models with a balanced dataset helps pay equal attention to all classes but may cause the model to focus too much on random variations (noise) within those classes. On the other hand, using an imbalanced dataset could result in not learning enough about the less prevalent class. Yet, it might lead to a simpler model that works better overall, particularly if the more common class reflects the general trends in the data. This analysis shows that experimenting with the ratios of classes might result in higher accuracy and, therefore, should be part of the model training phase.

Another important structural consideration is phishing by industry. Cybercriminals don't target all industries equally. They tend to focus on some businesses more than others. A summary of the share of phishing by industry can be seen in Table 11. This analysis was conducted on quarterly reports from APWG (similar to [6] and [7]) for the last five years. As can be seen, over the years, phishing against certain industries has dropped (Saas/Webmail), while for others, it has increased significantly (social media, logistics, shipping).

The most consistent and high figures are linked to companies in the finance domain (Financial institutions and Payments). If the phishing data in the dataset were collected from multiple sources or a single source with sufficient market coverage and during a long enough period, phishing records would have a similar distribution of impacted industries. Ensure that the creation of training and validation datasets contains a sufficient sample of the phishing attack against various industries. With the legitimate data, the distribution of collected records doesn't have to copy the distribution of phishing pages as per Table 11, but since the phishing record will, it is important to represent the legitimate pages from the most targeted industries sufficiently. This will provide pattern variability for the model to distinguish phishing from legitimate industry pages.

The above structural considerations are the most common ones, but others might be relevant and depend on your particular use case. One such example might be URL shorteners. Phishing records will most likely contain URL shorteners as they are quite common, with

occurrence between 0.2% and 0,7% [37]. So, out of each 1,000 phishing records, there will be between 2 and 7 phishing records with URL shorteners. If the dataset contains only legitimate webpages with an actual domain in the URL, whereas there will be phishing records using shorteners, such structural imbalance could impact the model's accuracy as the model will only see phishing records with URL shorteners.

Table 11 Average share of phishing per industry per Year.

	2019	2020	2021	2022	2023	Average
SaaS/Webmail	34%	30%	19%	19%	18%	24.7%
Financial inst.	18%	20%	24%	25%	24%	22.0%
Other	14%	11%	10%	19%	11%	13.1%
Payment	22%	13%	9%	5%	6%	11.8%
Social Media	2%	11%	14%	12%	20%	11.2%
Retail/e-comm	4%	7%	12%	7%	5%	7.3%
Logistics/Shipping	1%	4%	5%	6%	7%	4.1%
Telecom	2%	1%	1%	2%	6%	2.3%
Crypto	0%	0%	5%	4%	2%	2.3%
Cloud/File Host	3%	2%	0%	0%	0%	0.2%
Gaming	0%	0%	0%	0%	2%	0.2%
Government	0%	0%	0%	0%	1%	0.1%

* The "Average" column is calculated as a mean value across all five years

An important consideration impacting the efficacy of the phishing detection model is source data language variability. Given phishing's global reach, a dataset enriched with multilingual content will strengthen the model's ability to discern phishing attempts across various languages, enhancing detection accuracy. Combining webpages in multiple languages eliminates linguistic biases and assures robustness against phishing strategies exploiting language-specific variations.

An example of how important it is to use the recent data for training the model, which should be used in real-world deployment, is the addition of new g-TLD domains (.dad, .phd, .prof, .esq, .foo, .zip, .mov, .nexus) that happened in the first half of 2023. The domain ''.zip'' captured the highest interest of security researchers as it perfectly mimics the .zip archive extension, which can be easily used for phishing purposes. When we ran a search within the PhishTank and PhishStats records from 2023, we found already more than 40 unique URLs with the new gTLDs reported as phishing (e.g., url.zip, newdocument.zip, microsoft-office.zip, tax-return-2022.zip, irsrefund.zip, etc.)

3.3.2 Creation Of The Current Dataset For Phishing Detection

Following the framework's steps, we created two datasets for legitimate data – one containing 100,000 records and another one containing 30,000 records, all of which were randomly selected from Common Crawl's index files (snapshot of CC_2023_50). These datasets don't have any overlapping records as their content is chosen from a bigger pool (approx. 250,000 records). This way, we prepared sufficient data for the training and testing models.

We selected complete data from PhishTank from the year 2023 for phishing records. We decided to use only PhishTank data as people manually review and flag them as phishing. PhishStats and OpenPhish (to the best of our knowledge and indications through the provided columns) use machine learning algorithms for evaluating phishing URLs, which could introduce an unnecessary "noise" in the form of incorrectly classified records (FP – legitimate records being phishing), especially when we don't really know the accuracy with which those models work – as it is not published or publicly known.

3.3.3 Data Transformation And Cleansing

Any collected data, especially those gathered through web scrapping (including cutting out the substring from within the predefined HTML tags), require a data quality review, which usually starts with general profiling and, if needed, a data transformation and/or data cleansing and filtering.

The goal of this step was to evaluate the quality of the data – the content, the variables type, distribution of the values and identification of potential lookup values, review of the lengths of the character variables, and the possible need to adjust the extraction scripts within the data capture application, deduplication of the data and removal of outliers or incomplete data. As part of this step, we also cleansed the data from values that might have been a typo, a result of incorrect submission by the user, or other reasons. Such data would add "noise" or values of features, which could decrease the model's accuracy.

The first filtering that was applied to the data was the selection of only confirmed phishing. This step is specific to PhishTank and wouldn't be needed for data from PhishStats or OpenPhish as they only share confirmed phishing data (though OpenPhish provides an

updated dataset for false positives that were reported as phishing and later were cleared as legitimate URLs). This step reduced the dataset from ≈417,000 records to ≈252,000 records. The next step - also described in the framework - was records de-duplication. After deduplication, which removed all records that had the same domain (we use five domain levels of granularity, meaning the five highest domains from the overall domain used in the URL – example URL in Figure 21 has exactly five domain levels) and were reported within 24 hours from the first reported instance, dataset shrank to ≈201,000 records.

3.3.4 Data Enrichment And Features Engineering

The next step was the preparation of the features/characteristics that would be used to distinguish the legitimate pages from the phishing ones.

As described in Chapter *2.2.4*Phishing Webpage Indicators (Characteristics) we proceeded to derive features from several areas:

- URL-based features - we derived 155 URL-based characteristics from 7 sub-components of the URL – URL itself, scheme, authority, path, filename, query, fragment - including directly indicating the presence of the URL's domain in our master Blacklist and Greylist created from all three sources (PhishTank, PhishStats, and OpenPhish) across all historical periods. Details of all features – names and their description are available in ***Appendix 1 – URL-based Characteristics.***
- HTML-based features – we derived 84 features that revolve around various characteristics linked to the content of the HTML webpage. Details of all features – names and their description are available in ***Appendix 2 – HTML-based Characteristics.***
- 3rd party data-based features (whois, ping, ip-geo location) – we derived 14 features using 3rd party information linked to the domain. The majority – 9 features – are linked to whois and primarily to domain registration, expiration, and update dates. Two features are related to ping details and three features are linked to IP-geo location. Details of all features – names and their description are available in ***Appendix 3 – 3rd Party-based Characteristics.***

We derived 253 features available for both datasets with legitimate records (100K and 30K). The PhishTank dataset was slightly different as we didn't have 100% details for all

records. Therefore, from the phishing data, we created several datasets with different set of features:

- PhishTank dataset with 155x URL-based features with **≈201,000 records**
- PhishTank dataset with 155x URL-based features and 84x HTML-based features with **≈140,000 records** (this is the result of the short lifespan of phishing webpages as described in Chapter ***3.2.3.4***)
- PhishTank dataset with 155x URL-based features, 84x HTML-based features, and 9x whois-based features with **≈47,000 records**
- PhishTank dataset with 155x URL-based features, 84x HTML-based features, 9x whois-based and 2x ping-based features with **≈47,000 records**
- PhishTank dataset with 155x URL-based features, 84x HTML-based features, 9x whois-based features, 2x ping-based features and 3x IP-Geo location-based features with **≈8,000 records**

3.4 Models Training And Validation

During this phase, we tried to answer pressing questions about creating the best possible model for detecting phishing web pages. We conducted several experiments that helped us identify the most perspective model algorithm, minimum set of features, and potential areas of improvement in achieving the highest possible accuracy measures.

3.4.1 Incremental Value Of Additional Features

We spent a lot of our research time on data collection. We can confirm that most efforts involved several cycles of identification, gathering, storing, manipulating, and reviewing the data. Therefore, it is not surprising that the first question we wanted to get answered was – ***Are the additional data from various domains contributing to the model's accuracy and, if so, how much?*** One of the fundamental facts related to predictive modeling is that the quality of the model is directly derived from the data quality and is heavily dependent on the availability of relevant indicators that allow for classification into the defined sub-classes. We were able to collate an extensive set of various characteristics from multiple areas (url,

domain, HTML webpage, etc.), which should aid the model's training to find the most indicative features and build a robust and accurate phishing detection model.

As we had various volumes of data with various extend of the features, we selected the size of the training and testing dataset in such a way that we would be able to run with the same configuration (dataset sizes, ratio of phishing and legitimate records) across various scenarios. Our experiments were designed to evaluate selected KPIs leveraging a training dataset with 6,000 records, testing dataset 2,000 records and ratio of phishing vs. legitimate records was 1:1.

With this setup, we ran the following scenarios (features were incremented):

1. Selected algorithm with only 155x URL-based features (**U**)
2. Selected algorithm with additional 84x HTML-based features (**UH**)
3. Selected algorithm with additional 9x Whois-based features (**UHW**)
4. Selected algorithm with additional 2x Ping-based features (**UHWP**)
5. Selected algorithm with additional 3x IP-Geo-based features (**UHWPG**)

The results collected from executing all scenarios are summarized in Table 12. The first column of the table identifies the algorithm that was trained on 6,000 records of training data and tested on 2,000 records of testing data. These volumes are also provided in the second column. The third column captures the number of seconds it took to train and test the model. Each of the following five columns represents the extent of features available within the dataset as described in the list above this paragraph. Each row represents the collected Balanced Accuracy measure for a combination of the given algorithm(row) and features set(column).

We have trained and evaluated the performance of the following models:

- Logistic Regression model (LR)
- Decision Tree (DTree)
- Support Vector Machines (SVM)
- Random Forest (RF)
- K-Nearest Neighbor (KNN)
- Naïve Bayes (NB)
- Gradient Boost (GBoost)

- Adaptive Boosting (ABoost)

Table 12 Balanced Accuracy across various algorithms and set of features

Model	Data	Dur.	U	UH	UHW	UHWP	UHWPG
Abr.	Obs	s	BAcc	BAcc	BAcc	BAcc	BAcc
LR	6000	14	0.949	0.962	0.977	0.978	0.974
LR	**2000**	2	**0.946**	**0.955**	**0.976**	**0.983**	**0.973**
Dtree	6000	25	0.949	0.961	0.956	0.963	0.960
DTree	**2000**	4	**0.954**	**0.956**	**0.967**	**0.972**	**0.969**
SVM	6000	251	0.947	0.960	0.977	0.977	0.974
SVM	**2000**	46	**0.947**	**0.956**	**0.974**	**0.982**	**0.972**
RF	6000	137	0.970	0.978	0.981	0.982	0.981
RF	**2000**	30	**0.972**	**0.974**	**0.982**	**0.986**	**0.988**
KNN	6000	16	0.933	0.940	0.966	0.967	0.958
KNN	**2000**	4	**0.939**	**0.941**	**0.964**	**0.972**	**0.964**
NB	6000	6	0.743	0.569	0.528	0.525	0.626
NB	**2000**	2	**0.736**	**0.557**	**0.527**	**0.526**	**0.629**
GBoost	6000	494	0.970	0.979	0.981	0.982	0.983
GBoost	**2000**	86	**0.976**	**0.980**	**0.986**	**0.986**	**0.982**
ABoost	6000	286	0.962	0.980	0.977	0.980	0.979
ABoost	**2000**	62	**0.965**	**0.974**	**0.982**	**0.986**	**0.978**

The above table answers whether additional features from new areas provide incremental value, and if yes, how much? They do. Adding extra features from various areas allows the model to reach better accuracy. This could be observed for all models except Naïve Bayes, which showed mixed results and surprisingly deteriorated the more data we used for training (we did an experiment where we used data of various sizes, and the best Accuracy was achieved with the lowest volumes - 600 records - and as the data grew the accuracy decreased. There could be multiple reasons for such behavior, but the most relevant ones identified were a) used features are not independent – which is undoubtedly true for features derived from the domain and based on HTML content of the landing page, b) Naïve Bayes performs relatively poorly when there are irrelevant or less predictive features, which in our case was certainly the case.

The best-performing models in this experiment, availing hundreds of features, were Random Forest, Gradient Boosting, and Adaptive Boosting models. These three models not only achieved the highest Balanced Accuracy on the dataset with a complete set of features

(253) but also achieved the best results on the dataset with a lesser set of features, ≈98% Balanced Accuracy on a dataset with UR and HTML-based features (UH).

3.4.2 Best-performing Features

Another question that was closely related to the deployment of the trained model into our solution to detect phishing in real-time was: ***What minimum number of features with maximum impact on the model shall we use?*** Having an answer to this question has helped to define the scope of features that we would deploy into our phishing detection solution. Here, the motivation is more towards as few features as practical because every additional feature creates additional complexity to deploy the model and pre-calculate or derive the features necessary for the model to assess the reported URLs.

We used a Recursive Feature Elimination (RFE) method to identify features that contribute the most to predicting the target variable – in our case, the URL being phishing. In simple terms, RFE repeatedly constructs a model and chooses either the best or worst performing feature, sets it aside (best features) or cuts them away (worst performing), and then repeats the process with the rest of the features until all features in the dataset are exhausted or the specified number of features is selected/left. In our case, we defined five buckets of sizes – 5, 10, 15, 20, and 25 features. We used a Logistic regression algorithm for feature selection as it can provide a measure of importance for each feature.

The RFE method was applied on the same dataset used (6,000 records in the training dataset and 2,000 records in the testing dataset) in Chapter 3.4.1 so that it is easy to compare the Balanced Accuracy figures and how much or how little it decreased compared to models trained on complete sets of features for each area(U with 155 features, UH with 155+84 features, etc.).

The resulting sets of best-performing features are:

- **5 features list:** aut_www_flg, who_expiry_soon_flg, aut_blacklist_d3_flg, who_hirisk_iana_flg, htm_href_susp_flg
- **10 features list:** aut_www_flg, aut_hirisk_tld_flg, who_hirisk_iana_flg, aut_tld_top10p_flg, aut_urlshort_flg, fnm_pdf_flg, who_expiry_soon_flg, aut_hirisk_30brand_cnt, htm_href_susp_flg, aut_blacklist_d3_flg

- **15 features list:** aut_hirisk_tld_flg, aut_blacklist_d4_flg, aut_hirisk_30brand_cnt, url_space_cnt, aut_greylist_d3_flg, aut_blacklist_d5_flg, aut_tld_top10p_flg, who_expiry_soon_flg, aut_blacklist_d3_flg, aut_avg_domain_len, who_hirisk_iana_flg, aut_urlshort_flg, url_upcase_pct, aut_www_flg, htm_href_susp_flg
- **20 features list:** aut_blacklist_d4_flg, who_expiry_soon_flg, url_space_cnt, url_upcase_pct, aut_dom_cnt, aut_urlshort_flg, aut_hirisk_tld_flg, who_reg_less_1y_flg, aut_greylist_d3_flg, htm_href_susp_flg, aut_tld_top10p_flg, aut_blacklist_d5_flg, fnm_pdf_flg, who_hirisk_iana_flg, who_recent_update_flg, aut_avg_domain_len, aut_blacklist_d3_flg, aut_www_flg, sch_http_flg, aut_hirisk_30brand_cnt
- **25 features list:** aut_dot_cnt, htm_href_susp_flg, htm_wrd_trademark_cnt, sch_http_flg, aut_hirisk_tld_flg, htm_input_password_cnt, who_hirisk_iana_flg, aut_tld_top10p_flg, aut_blacklist_d4_flg, aut_blacklist_d5_flg, aut_www_flg, aut_blacklist_d3_flg, fnm_hyphen_cnt, who_reg_less_1y_flg, aut_greylist_d3_flg, who_expiry_soon_flg, aut_hirisk_keyword_cnt, htm_form_get_cnt, aut_urlshort_flg, who_recent_update_flg, url_space_cnt, htm_audio_video_cnt, aut_hirisk_30brand_cnt, aut_avg_domain_len, url_upcase_pct

The feature-wise listing of all sets of best-performing features is aligned in ***Appendix 5 – Best Performing Features Comparison Table.*** The results collected from the execution of all scenarios are summarized in Table 13. The first column of the table identifies the algorithm, which was trained on 6,000 records of training data and tested on 2,000 records of testing data. These volumes are also provided in the second column. Each of the next five columns represents the extent of features available within the dataset as described in the list above this paragraph. The first column has selected 5 best-performing features as per RFE, the second has 10 best-performing features as per RFE, etc. Each row represents the collected Balanced Accuracy measure for the combination of a given algorithm(row) and features set(column).

The improvement between the Top5 and Top10 columns means adding 5 columns to existing 5 columns to those already in the Top5 dataset - the model improves on average by

2%. The average improvement between Top10 and Top15 columns, adding 5 more columns, is 4%, the biggest improvement achieved between two buckets with top-performing features. The Top20 column improved by 1.4% on average, and the dataset with Top25 features on average didn't improve at all; it achieved 0% average improvement. If we ignored the degradation in the Naïve Bayes row, we would get a 0.7% improvement to the previous Top20 column. But this last column already showcases the rule of diminishing returns, whereby adding 5 more columns, the incremental improvement decreases.

Table 13 Balanced Accuracy of algorithms and sets of best-performing features

Model	Data	Top5	Top10	Top15	Top20	Top25a
Abr.	Obs	BAcc	BAcc	BAcc	BAcc	BAcc
LR	6000	0.906	0.910	0.959	0.967	0.969
LR	**2000**	**0.904**	**0.908**	**0.957**	**0.964**	**0.965**
Dtree	6000	0.906	0.910	0.956	0.959	0.960
DTree	**2000**	**0.904**	**0.908**	**0.956**	**0.960**	**0.965**
SVM	6000	0.904	0.908	0.959	0.966	0.968
SVM	**2000**	**0.904**	**0.908**	**0.956**	**0.964**	**0.966**
RF	6000	0.906	0.910	0.960	0.966	0.967
RF	**2000**	**0.904**	**0.909**	**0.958**	**0.966**	**0.972**
KNN	6000	0.840	0.860	0.954	0.959	0.961
KNN	**2000**	**0.819**	**0.907**	**0.958**	**0.962**	**0.964**
NB	6000	0.823	0.897	0.868	0.927	0.906
NB	**2000**	**0.819**	**0.907**	**0.870**	**0.940**	**0.911**
GBoost	6000	0.906	0.909	0.959	0.968	0.970
GBoost	**2000**	**0.904**	**0.908**	**0.956**	**0.965**	**0.974**
ABoost	6000	0.902	0.904	0.944	0.952	0.956
ABoost	**2000**	**0.900**	**0.903**	**0.939**	**0.952**	**0.962**

The best-performing models in this experiment trained on the selection of best-performing features were the Random Forest model, with testing Balanced Accuracy equal to 0.972, and the Gradient Boosting model, with 0.974.

3.4.3 Incremental Value Of Additional Data

When we analyzed all the best-performing features in all buckets, we confirmed that there was no feature from Ping (2 features) or IP-Geolocation (3 features) areas in our selection. This presents an opportunity to train and test the model on a much bigger dataset,

as the primary constraint for the size of the dataset data was due to the lack of IP geolocation details (we had only a little more than 8,000 records with all these details present).

This experiment evaluates ***whether training on more data than 6,000 using the selected 25 best-performing features will improve the model's accuracy and, if so, by how much***. We tested two different versions of Top25 features – those derived from the training dataset containing 6,000 records, which we named Top25a, and a new set of 25 best-performing features derived from a new training dataset with 50,000 records, which we named Top25b.

By applying the RFE method to our new training dataset to identify 25 best-performing features, we ended up with these:

- **URL:** url_upcase_pct, url_len, url_space_cnt, url_questionmark_cnt
- **SCH:** sch_http_flg, sch_secure_flg
- **AUT:** aut_www_flg, aut_blacklist_d3_flg, aut_avg_domain_len, aut_blacklist_d5_flg, aut_hirisk_keyword_cnt, aut_hirisk_30brand_cnt, aut_dot_cnt, aut_dom_cnt, aut_len
- **PTH:** pth_len, pth_dir_cnt, pth_dot_cnt, pth_space_cnt
- **FNM:** fnm_len
- **QRY:** qry_len
- **FGM:** fgm_token_cnt, fgm_len
- **HTM:** htm_href_susp_flg
- **WHO:** who_expiry_soon_flg

At this point, we had two lists of Top25 best-performing features – Top25a (derived from the 6,000 records training dataset) and Top25b (derived from the 50,000 records training dataset). As a pre-requisite to run this experiment, we prepared 2 versions of the training (50,000 records) and testing (10,000 records) dataset in the next steps. The first version of the datasets only contained columns listed as Top25a, and the second version contained only columns listed as Top25b. The experiment consisted of training various algorithms on the 50,000 records dataset with respective columns and evaluating the models on the testing dataset with 10,000 records.

The results collected from the execution of the scenarios are summarized in Table 14. The first column of the table identifies the algorithm that was trained on 50,000 records of training data and tested on 10,000 records of training data. These volumes are also provided

in the second column. The next two columns represent two different sets of Top25 features available within the dataset. Each row represents the collected Balanced Accuracy measure for the combination of a given algorithm(row) and features set(column).

Table 14 Balanced Accuracy of algorithms and sets of 25 best-performing features

Model	Data	Top25a	Top25b
Abr.	Obs	BAcc	BAcc
LR	50000	0.967	0.962
LR	**10000**	**0.967**	**0.962**
Dtree	50000	0.973	0.967
DTree	**10000**	**0.973**	**0.966**
SVM	50000	0.967	0.966
SVM	**10000**	**0.966**	**0.966**
RF	50000	0.975	0.978
RF	**10000**	**0.975**	**0.977**
KNN	50000	0.972	0.971
KNN	**10000**	**0.972**	**0.969**
NB	50000	0.813	0.757
NB	**10000**	**0.821**	**0.751**
GBoost	50000	0.973	0.971
GBoost	**10000**	**0.971**	**0.970**
ABoost	50000	0.961	0.959
ABoost	**10000**	**0.958**	**0.957**

The first question for which we were trying to get an answer was – ***By increasing the volume of records while keeping the list of features fixed, do we get to train a more accurate model?*** Now, looking at Table 14, column Top25a, and comparing it to Table 13, column Top25a, and focusing only on performance achieved on testing data (ignoring Naïve Bayes results as those introduce a lot of erratic figures and complicate comparison), we can calculate that the average improvement across all models is 0.2%. By increasing the volume of records for training the model from 6,000 to 50,000, we gained a 0.2% average improvement. However, when we look at individual algorithms, two algorithms - Decision Tree and KNN - stand out, both improving by 0.8%.

Another question we tried to answer with this experiment is – ***Will the list of best-performing features change if we gather more data while the list of features remains the same?*** The answer is yes, as we described the content of both lists, Top25a and Top25b. These two lists share 12 out of 25 columns. The remaining 13 columns are different, and this

difference results from having significantly more observations in the dataset from which the features were selected, even though the list of features was the same. But was the list of features the same? Not entirely! List Top25a was derived from 253 columns and 6,000 records dataset. Top25b was derived from 248 columns (5 columns related to Ping and IP-Geo location were removed). But could these missing columns - none of which were selected as best-performing in Top25a - impact the selection of the Top25b? Though the probability of them impacting the selection process is rather small, we can't entirely diminish the potential impact on the selected variables due to unknown correlations or collinear relationships between these removed columns and other columns that stayed in the dataset. However, it is more probable that the impact of differences between the selected features is due to the size of the dataset, whereas the more extensive datasets have greater statistical power, which can reduce the influence of noise and outliers. This might lead to a more accurate representation of features' importance and make previously non-apparent relationships and interactions between features more detectable.

The last question we wanted to answer with this experiment was – ***Having different sets of features but derived from the same data, will the balanced accuracy of the models be the same or different, and how different will they be?*** When we calculate an average Balanced Accuracy across various models (again ignoring the results of the Naïve Bayes algorithm) for both testing datasets Top25a and Top25b, we can see that the figure is almost the same: 0.969 vs. 0.966. Interestingly, models trained on the features identified on the smaller dataset (6,000 records) and applied on the bigger dataset (50,000) achieved marginally better Balanced Accuracy than the model using new features derived from the bigger dataset (50,000 records). However, it is important to note that the balanced accuracy difference is minimal, only 0.03.

The best-performing models in this experiment with datasets having only 25 selected features were again - Random Forest, Gradient Boosting, and, in this experiment, also KNN models. For the dataset built with Top25a features, the decision tree also achieved very close performance metrics.

3.5 Implementation Of Phishing Detection – PhishCheck

PhishCheck is a web application built to evaluate whether the provided URL is phishing or legitimate in real-time. The user interface is built using PHP. Flask, as a micro web framework written in Python, hosts the predictive model – in our case, the Random Forest model, which assesses the URL and is made available to PHP via the REST API layer built in Flask. High-level logical architecture is depicted in Figure 38.

PhishCheck, in its current version, has implemented two workflows:

a) Assessment of recently received reported URLs

b) Assessment of manually typed URL

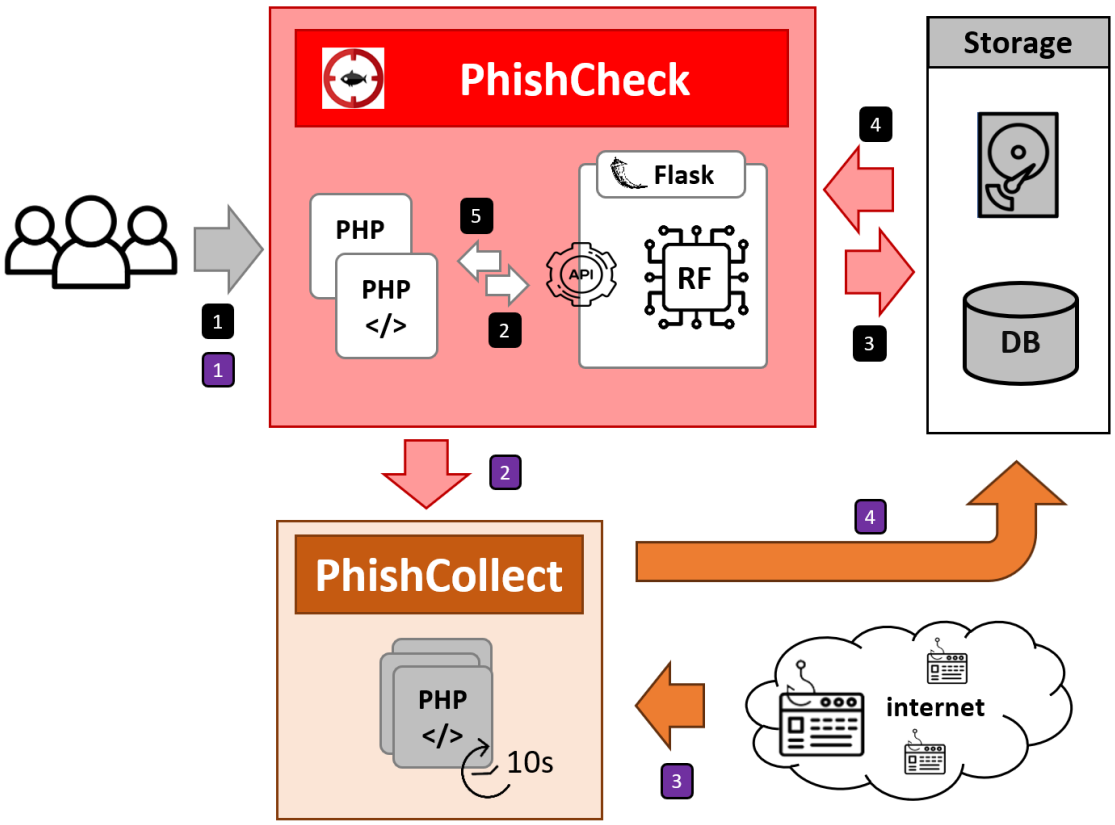


Figure 38 High-level logical architecture of PhishCheck

PhishCheck has deployed a Random Forest model with 30 best-performing features derived from URL, HTML, and whois with Balanced Accuracy at 97.7%. The final list of features used in PhishCheck can be seen in ***Appendix 5 – Best Performing Features Comparison Table*** in the last column - Top30.

3.5.1 Assessment Of Recently Reported URLs

In this flow (depicted in Figure 38 with steps numbered in **black squares**), the user can select from the recently reported URLs that the PhishCollect processed. The interaction starts by listing the recent URLs from our data feed (PhishTank, PhishStats, and OpenPhish), as shown in Figure 39.

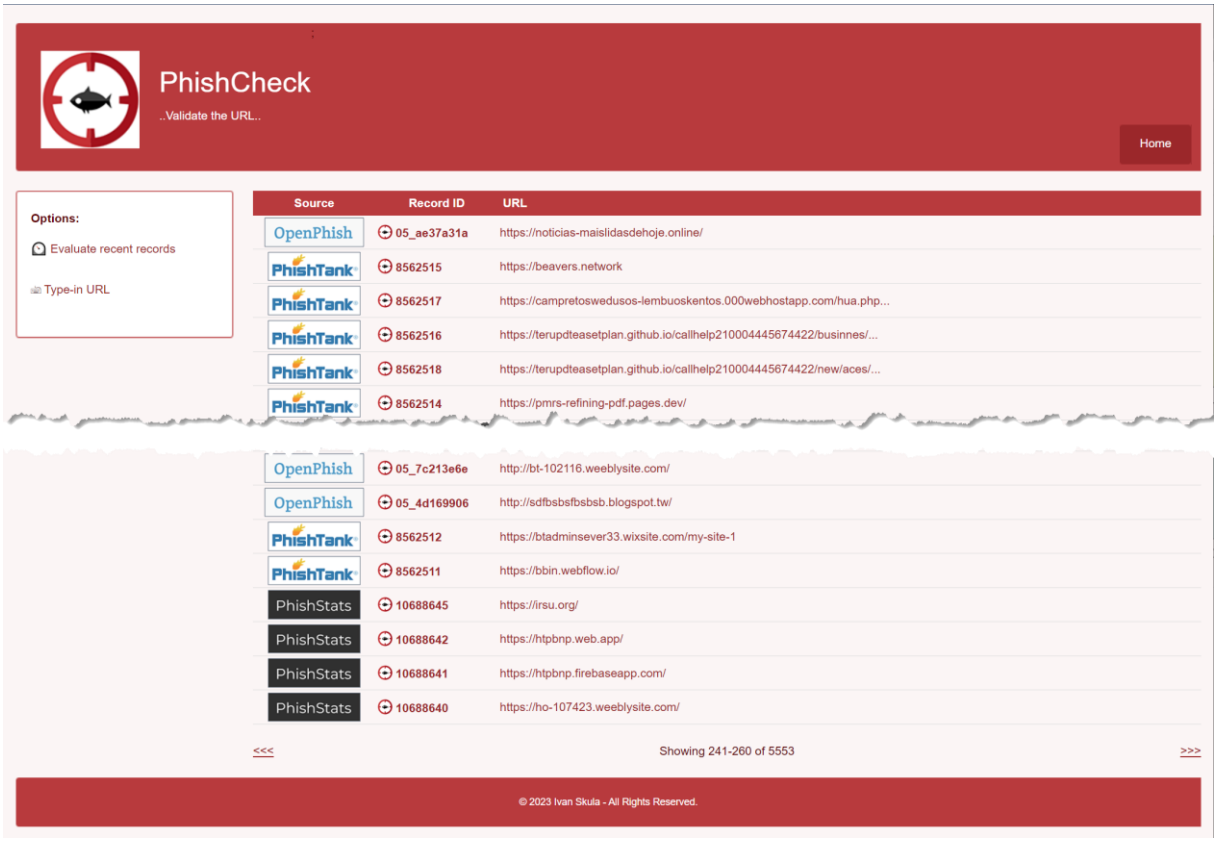


Figure 39 PhishCheck listing of the recently reported URLs

The users then - as highlighted in Figure 38 in point [1] - select from the listed URLs the one they want to assess by our deployed predictive model. In step [2], the selection is passed via API onto the Flask application. In step [3], the flask application will then reach into the database to collect the whois details and onto disk to collect the HTML page of the selected URL. These returned details in step [4] are used to derive the features required by the deployed predictive model. The predictive model evaluates the provided inputs and returns the response as JSON in step [5]. This response is formatted and presented by PHP to the user.

3.5.2 Assessment Of Manually Typed URLs

In this flow (depicted in Figure 38 with steps numbered in **purple squares**), the user provides the URL by manually typing it into the entry form in step [1]. Since this URL is not picked from the already processed URLs, we can't follow the same flow as no HTML and whois details are captured for it. Therefore, the manually entered URL is first passed to PhishCollect for processing. PhishCollect – as depicted in steps number [3] and [4] in the purple square, will gather the details for the provided URL and save them into the DB and storage. Now, the user is re-directed to the previous(black) flow to the step [2]. This means the manually typed URL is passed onto Flask, and from here, the process continues with the remaining steps – [3], [4], and [5] as described in the previous flow.

3.6 Implementation Of Blacklist And Greylist

As can be seen in the complete list of features derived from the authority domain in ***Appendix 1 – URL-based Characteristics***, we defined six features that are directly derived from Blacklists and Greylists that we created. The process of creation of these lists is depicted in Figure 15. The only differentiator is the number of levels being used. In most of our research, we use the granularity of five levels (see Figure 21), but for practical application, we also build the versions using 4 and 3 levels of domains.

Features linked to the Blacklists and Greylists:

- aut_blacklist_d3_flg - D3 form of the domain of the URL is in our Blacklist
- aut_blacklist_d4_flg - D4 form of the domain of the URL is in our Blacklist
- aut_blacklist_d5_flg - D5 form of the domain of the URL is in our Blacklist
- aut_greylist_d3_flg - D3 form of the domain of the URL is in our Greylist
- aut_greylist_d4_flg - D4 form of the domain of the URL is in our Greylist
- aut_greylist_d5_flg - D5 form of the domain of the URL is in our Greylist

Blacklist and Greylists are built from all the data feeds – PhishTank, PhishStats, and OpenPhish throughout the whole available period. And as we can see from the experiments selecting the best-performing feature, some of the Blacklist and Greylist-linked features are present among the top-performing features.

Conclusion

Detecting phishing webpages is not an easy task, not for humans or computers. Attackers who build phishing webpages are incredibly adaptable and agile. They try every new angle, technology, and approach to improve their odds. Phishing is also a tool of a wide variety of attackers – from opportunistic and not very technically savvy to those who perform cyber-attacks as part of their job or primary source of income. Some attacks are prepared in great detail and tuned to the carefully selected recipient, while others are mass-produced following the common patterns that were already proven to work. Phishing is a multi-faceted problem – and all these various facets we summarized in this work. It is extremely important to recognize and be aware of the many forms a phishing attack can take, as without this knowledge, building an actual phishing detection solution would be hard, if not impossible. The breakdown of phishing to its building blocks – channels, forms, use cases, data, distinguishing characteristics, etc.- makes up the foundation based on which the design of our solution (or any solution for that matter) stands.

Another layer contributing to our solution's final design and quality is built around a few selected branches of research, which aimed at exploring how to develop and improve on techniques employed by other researchers or validate our own hypothesis. One such research focused on the relevancy of using Blacklist and Greylist, especially in the recent years. This research confirmed our original assumption about diminishing the direct effect of the Blacklist – in 2022, only 6.1% of reported domains were re-occurring. Therefore, this is the maximum potential contribution of the Blacklist to detect phishing. Despite this low figure – a result of the shrinking trend of re-occurring domains – we believe that this low number is actually a result of the continuous use of Blacklists and a reason why the attackers have to register all those new domains. Another research we conducted and published focused on the prevalence of common URL obfuscation techniques, which are still being used in the newly published research articles as an excellent indicator of phishing. Revelation - how rare these techniques are among the phishing URLs was quite surprising. In our research, we analyzed and documented the prevalence of 7 obfuscation techniques, and the final figure was that less than 3% of all phishing URLs employed at least one obfuscation technique. Therefore, detection relying on these indicators would have minimal success and coverage. Another

detailed analysis focused on the phishing webpages' longevity (or lifespan). The main observation of this research is confirmation of the very short longevity of phishing webpages, where 35.1% of reported phishing webpages are unavailable immediately after they are made available to the most common phishing listings – PhishTank and OpenPhish. After the first five minutes, an additional ≈9% of reported ULRs become unavailable; after 24 hours, only ≈41% of URLs are available, and after 48 hours, only 36% of reported URLs remain active. Anyone planning to collect the phishing data must access and capture all the details as soon as the URL is reported and made available. Any minute delay – especially immediately after the URL is reported, results in a sharp decrease in the chance to gather the details and shrinks the pool of relevant data.

All of the theoretical objectives of the thesis, as listed in the Introduction chapter, along with the published research briefly described above, have a material and actual impact not only on the practical goal of our thesis – our phishing detection solution – but also on other possible branches of research. One of the observed problems in the current study of phishing is the limited possibility of comparing different researchers' results. Though they might have used the same source of data, they often transform the data in such a manner that makes further comparison of the results impossible or at least very limited. Many don't provide sufficient details about the data source or the period when the data were collected. Adjustments to the underlying collected data, like adding genuine pages to the collected phishing data, different data cleansing techniques for duplicate data, outliers' removal, or even data imputation, are just a few examples of actions that negatively impact the comparability of the research outcomes, if not captured, documented and transparently communicated. A standardization or unification framework for steps preparing the dataset, which we created and published, should help provide guidelines to aid researchers with this task. Another hands-on contribution to the research in this area is captured in the data-related chapters – a summary of problems encountered during our data collection process. Phishing webpage longevity is one of the observed problems, as was already mentioned. Summarizing the others – country-level filtering, the impact of antivirus, or anti-scrapping technologies – along with the suggested solutions provides practical help to other researchers. Also, the overall design patterns and considerations related to data acquisition can be generalized and reused for any web-scraping project. And finally, the main contribution is to designing and

implementing real-time phishing detection solutions. As part of this process, we performed a variety of experiments trying to identify a) the most relevant indicators to distinguish legitimate URL/webpage from phishing from among 253 features, b) the best performing (accuracy of detection) algorithm from the pool of 8 and also c) optimize the size of the dataset to achieve the most accurate model.

During our research, we tried to pay close attention to any potential limitations that could hamper or skew the results or findings. To eliminate potential data dependency in our study, we tried to build the datasets using multiple sources. And though we deployed the model trained on data only from PhishTank, this was a conscious decision to get a baseline model and baseline KPI figures – as we believe that the data from PhishTank are most accurate due to the human-driven review and phishing tagging process. The data we collected and used in our research should contain a sufficient variety of phishing attacks as it is sourced from multiple sources. The same applied to legitimate data – sourced and laboriously processed from the best possible representation of the web – Common Crawl. Also, the majority of the data (from 2022 onwards) were collected immediately as they were reported, so we should have all the existing variety of reported phishing webpages that were possible to capture. Nevertheless, there might be certain types of phishing attacks that are rare or have extremely short lifespans, which might be under-represented within the collected data (e.g. within the 35% of phishing URLs we observe as inactive when they are made available in the data feed). Regarding potential model overfitting, we leveraged a separate testing dataset of sufficient size and clearly reported the selected KPIs for the training and testing datasets separately. Looking at various published papers and the techniques they used to detect phishing, the selection of algorithms we evaluated could be considered a limitation. Yet again, this was a conscious decision where we considered our set of selected algorithms as a baseline, and we plan to evaluate and further improve the solution by evaluating other – more complex algorithms or their combinations. Regarding the features, we tried to deploy as many as we could identify and implement. We used 253 features from various areas linked to the reported URL and webpage.

One limitation of which we are aware is the way the top-level domain (TLD) is formally defined and how it is used in practice in some countries. Some countries do not use only their ccTLD domain but often attach another subdomain to it – e.g., **co.uk**, **com.cn**, **com.au,** etc. In

all our research, we worked with the standard TLDs and ignored this nuance, but in practical terms – this approach reduced the ability to gather details based on registrable domain (normally SLD.TLD (e.g., sme.sk), but for these domains, the registrable domain is, in reality, THLD.SLD.TLD, e.g., https://**www.amazon.co.uk**/, TLD is "**uk,**" and the second-level domain is "**co**," but the registrable domain – domain which should be looked up is not **co.uk** but **amazon.co.uk** since the **co.uk** serves as an actual top-level domain. Considering this specific use-case would improve the accuracy and, if translated into the features, should also slightly improve the model's accuracy.

Many potential routes exist to extend, improve, or complement our work. On the data side of the research, we suggest a review and improvement of the collection process of details for phishing data as we observed a low volume of details captured for certain aspects (e.g., IP-geo location details, whois details, and ping details – all derived from registrable domain) by addressing the multi-level TLD as described in the limitations section. Additionally, we suggest optimizing the process of capturing the features as the current process stores most HTML-related details on the drive as files being archived. Extracting the features from these data requires a lot of data being manually copied to the server from the archive, extracted, and prepared for processing, including a non-negligible storage requirement and a requirement to maintain a separate process to extract the relevant details from these archived files.

As stated in the chapter describing the extent of data being collected, we propose an evaluation of additional features focused around favicon as well as analysis of the screenshot of the rendered webpage screenshot. The next area of focus could be an evaluation of other types of algorithms and techniques – mainly neural networks and potential multi-modal models utilizing text data along with images to spot phishing. An interesting yet very little researched area is the stability of the detection model throughout longer periods. Questions like - how long does the model perform within pre-defined quality thresholds? Which features are more volatile and causing the deterioration of the model, and which are more stable and still indicative? These are examples of research questions linked to this topic.

One high-priority branch of research could focus on brand identification, as most phishing webpages fall into this category. If we could accurately identify imitated brands, we could identify a significant portion of phishing with high accuracy simply by comparing the URL

with the homepage of those brands. Brand detection would allow us to create alerts and monitor phishing attempts of particular brands.

If done accurately and promptly, the detection of phishing webpages significantly impacts the users' experience when interacting with the online world and their sense of security. Such a solution mitigates the direct losses from phishing. In addition, it helps reduce the impact of many other, more complex cyber-attacks that utilize phishing for a specific purpose in a much larger scheme. Such an attack could be devastating and on a completely different scale (regional or even a country-wide impact). Despite the high interest in this area among researchers, there is no straightforward way to tackle the problem. Since phishing is a multi-billion-dollar problem, even a small contribution within this area might result in a meaningful impact on the lives of the common people – potential victims.

Bibliography

[1] M.A. Adebowale, K.T. Lwin, E. Sánchez and M.A. Hossain, "Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text," in *Expert Systems with Applications*, 2019, vol. 115, pp.300-313

[2] Akamai, *phishing - baiting the hook*. Accessed on: Apr. 2, 2022. [Online]. Available: https://www.technologie-onderwijs.nl/to/akamai/Akamai-soti-security-phishing-baiting-the-hook-report-2019.pdf

[3] W. Ali and A.A. Ahmed, "Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," in *IET Information Security*, Nov. 2019, vol. 13, no. 6, pp. 659-669.

[4] APWG, Phishing Attack Trends Report – January, 2004, Accessed on: Apr. 2, 2022. [Online]. Available: https://docs.apwg.org/reports/APWG.Phishing.Attack.Report.Jan2004.pdf

[5] APWG, Phishing Activity Trends Report – December, 2006, Accessed on: Apr. 2, 2022. [Online]. Available: https://docs.apwg.org/reports/apwg_report_december_2006.pdf

[6] APWG, Phishing Activity Trends Report – Q1/2019, Accessed on: Apr. 2, 2022. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2019.pdf

[7] APWG, Phishing Activity Trends Report – Q3/2020, Accessed on: Apr. 2, 2022. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q3_2020.pdf

[8] M. Babagoli, M.P. Aghababa and V. Solouk, "Heuristic nonlinear regression strategy for detecting phishing websites," in *Soft Computing*, 2018, vol.23, doi:10.1007/s00500-018-3084-2

[9] 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 (ACSW '20)*, 2020, pp. 1–11.

[10] D.G.N. Benítez-Mejía, A. Zacatenco-Santos, L.K.Toscano-Medina and G. Sánchez-Pérez, "HTTPS: a Phishing Attack in a Network," in *Proceedings of the 7th International Conference on Information Communication and Management ICICM*, Aug 2017, pp. 24–27.

[11] G. Bonnin and A. Boyer, "Higher Education and the Revolution of Learning Analytics", 2017. Accessed on: Apr. 2, 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03012475

[12] Checkpoint, *Check Point Press Releases - DHL Replaces Microsoft as Most Imitated Brand in Phishing Attempts in Q4 2021*. Accessed on: Apr. 2, 2022. [Online]. Available: https://www.checkpoint.com/press/2022/dhl-replaces-microsoft-as-most-imitated-brand-in-phishing-attempts-in-q4-2021/

[13] Cisco, *Acquisitions by Year*, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.cisco.com/c/en/us/about/corporate-strategy-office/acquisitions/acquisitions-list-years.html

[14] A. Das, S. Baki, A.E. Aassal, R.M. Verma and A. Dunbar, "SoK: A Comprehensive Reexamination of Phishing Research From the Security Perspective," in *IEEE Communications Surveys & Tutorials*, 2020, vol. 22, pp. 671-708.

[15] J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran and B. S. Bindhumadhava, "Phishing Website Classification and Detection Using Machine Learning," *2020 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2020, pp. 1-6, doi: 10.1109/ICCCI48352.2020.9104161.

[16] "*Domain name analysis*," Accessed on: Apr. 2, 2022. [Online]. Available: https://datagenetics.com/blog/march22012/

[17] DuoLab, *State of the Auth 2021*, Accessed on: Apr. 2, 2022. [Online]. Available: https://duo.com/blog/the-2021-state-of-the-auth-report-2fa-climbs-password-managers-biometrics-trend

[18] "*Email Usage Statistics 2024: How Many People Use Email?*," Accessed on: Mar. 29, 2024. [Online]. Available: https://wpdevshed.com/email-usage-statistics/

[19] F5, *2019 Phishing and Fraud Report*, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.f5.com/content/dam/f5-labs-v2/article/pdfs/F5Labs_2019_Phishing_and_Fraud_Report.pdf

[20] FBI Internet Crime Complaint Center, *2018 Internet Crime report*, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2018_IC3Report.pdf

[21] FBI Internet Crime Complaint Center, *2021 Internet Crime report*, Accessed on: Mar. 29, 2024. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2021_IC3Report.pdf

[22] FBI Internet Crime Complaint Center, *2023 Internet Crime report*, Accessed on: Mar. 29, 2024. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf

[23] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks." in *WORM '07: Proceedings of the 2007ACM workshop on Recurring malcode*, 2007, pp. 1–8.

[24] Government of U.K., *Cyber Security Breaches Survey 2021*, Mar 2021, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2021/cyber-security-breaches-survey-2021

[25] H.L. Gururaj and G. BoreGowda "Phishing website detection based on effective machine learning approach," in *Journal of Cyber Security Technology*., 2020, pp. 1-14.

[26] M. Islam and N. Chowdhury, "Phishing websites detection using machine learning based classification techniques," in *Proc. 1st Int. Conf. Adv. Inf. Commun. Technol*, 11 2016, pp. 1–4.

[27] A.K. Jain and B.B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," in *Enterprise Information Systems*, 2021, pp. 1-39.

[28] Kim Zetter, *How RSA got hacked*, Wired, Aug 2011, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.wired.com/2011/08/how-rsa-got-hacked/

[29] A.D. Kulkarni and L. Leonard, "Phishing Websites Detection using Machine Learning," in *International Journal of Advanced Computer Science and Applications*, 2019, doi:10.14569/IJACSA.2019.0100702

[30] V.S. Lakshmi, M.S. Vijaya, "Efficient prediction of phishing websites using supervised learning algorithms," in *Procedia Engineering, vol. 30, International Conference on Communication Technology and System Design 2011*, 2012, pp. 798–805.

[31] S. Mishra and D. Soni, "SMS Phishing and Mitigation Approaches," in *2019 Twelfth International Conference on Contemporary Computing (IC3)*, 2019, pp. 1-5.

[32] S.S.M. Motiur Rahman, T. Islam and M.I. Jabiullah," PhishStack: Evaluation of Stacked Generalization in Phishing URLs Detection" in Procedia Computer Science, 2020, vol. 167, pp. 2410-2418.

[33] ProofPoint, *2021 State of the Phish*, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.proofpoint.com/us/resources/threat-reports/state-of-phish-infographic

[34] M.A. Rader and S.M. Rahman, "Exploring Historical and Emerging Phishing Techniques and Mitigating the Associated Security Risks,"in International Journal of Network Security & Its Applications (IJNSA), July 2013, vol.5, no.4.

[35] Radicati group, *Email Statistics Report, 2018-2022*, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.radicati.com/wp/wp-content/uploads/2018/01/Email_Statistics_Report,_2018-2022_Executive_Summary.pdf

[36] R.S. Rao and A.R. Pais, "An Enhanced Blacklist Method to Detect Phishing Websites," in *Lecture Notes in Computer Science*, 2017, pp. 323–333. doi:10.1007/978-3-319-72598-7_20

[37] J. Reis, M. Amorim, N. Melão and P. Matos, "Digital Transformation: A Literature Review and Guidelines for Future Research," in *Trends and Advances in Information Systems and Technologies*, 2018, pp. 411–421.

[38] SAS, *Analytics - What it is and why it matters*, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.sas.com/en_us/insights/analytics/what-is-analytics.html

[39] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists", in *Proceeding of the Sixth Conference on Email and Anti-Spam, CEAS*, 2009

[40] Tessian, Spear Phishing Threat Landscape 2021, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.tessian.com/research/spear-phishing-threat-landscape/

[41] "The raise of social media", Accessed on: Apr. 2, 2022. [Online]. Available: https://ourworldindata.org/rise-of-social-media

[42] H. Tupsamudre, A. Singh, and S. Lodha, "Everything is in the name – a url based approach for phishing detection," in *Cyber Security Cryptography and Machine Learning; Springer: Cham, Switzerland*, 05 2019, pp. 231–248.

[43] Verisign, *THE DOMAIN NAME INDUSTRY BRIEF - Q3/2021 DATA AND ANALYSIS*. Accessed on: Apr. 2, 2022. [Online]. Available: https://www.verisign.com/assets/domain-name-report-Q12021.pdf

[44] Verizon, *2020 Data Breach Investigations Report*, Accessed on: Apr. 2, 2022. [Online]. Available: https://enterprise.verizon.com/resources/reports/2021/2021-data-breach-investigations-report.pdf

[45] Verizon, The history of Phishing, Accessed on: Apr. 2, 2022. [Online]. Available: https://www.verizon.com/business/en-be/resources/articles/the-history-of-phishing/

[46] W. Wang, F. Zhang, X. Luo and S. Zhang, "PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks," in *Security and Communication Networks*, 2019, pp. 1–15.

[47] P. Yang, G. Zhao and P. Zeng, "Phishing Website Detection based on Multidimensional Features driven by Deep Learning, "in *IEEE Access*, 2019, vol. 7, pp. 15196-15209.

[48] A. Zamir, H.U. Khan, T. Iqbal, N. Yousaf, F. Aslam, A. Anjum and M. Hamdani, "Phishing web site detection using diverse machine learning algorithms", in *The Electronic Library*, 2020, vol. 38 no. 1, pp. 65-80.

[49] Cambridge Dictionary, Accessed on: Apr. 2, 2022. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/phishing

[50] Wikipedia, Accessed on: Apr. 2, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Phishing

[51] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," 01 2009.

[52] T. Skybakmoen and V. Phatak, "Comparative Test Report - Q2 2021 Web Browser vs. Phishing" CyberRatings.org. [Online]. Available: https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RWLycn

[53] R. Howard, "Cyber Fraud: Tactics, Techniques and Procedures. ", Auerbach Publications, April 2009

[54] J. Ma, L. Saul, S. Savage, and G. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious urls," 06 2009, pp. 1245–1254

[55] P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION," Internet Requests for Comments, RFC Editor, RFC 1035, November 1987. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1035.txt

[56] G. Aaron, L. Chapin, D. Piscitello, and D. C. Strutt. Phishing Landscape 2022 - An Annual Study of the Scope and Distribution of Phishing. Interisle Consulting Group. [Online]. Available: https://interisle.net/PhishingLandscape2022.pdf

[57] T. Berners-Lee, R. Fielding T., and L. Masinter M, “Uniform Resource Identifier (URI): Generic Syntax,” Internet Requests for Comments, RFC Editor, RFC 3986, January 2005. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3986.txt

[58] R. Siedzik. (2001, April) Semantic Attacks – What’s in a URL? SANS Institute. [Online]. Available: https://www.giac.org/paper/gsec/650/semantic-attacks-url/101497

[59] R. Heartfield and G. Loukas, “A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks,” ACM Computing Surveys, vol. 48, pp. 1–39, 02 2016

[60] A. M. Costello, “Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA),” Internet Requests for Comments, RFC Editor, RFC 3492, March 2003. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3492.txt

[61] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, ‘‘Machine learning based phishing detection from URLs,’’ Expert Syst. Appl., vol. 117, pp. 345–357, Mar. 2019.

[62] J. Lee, P. Ye, R. Liu, D. M. Divakaran, and M. Chan, “Building robust phishing detection system: an empirical analysis,” in NDSS Symposium 2020, 02 2020.

[63] G. Vrbancic, I. Fister, and V. Podgorelec, “Datasets for phishing websites detection,” Data in Brief, vol. 33, p. 106438, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340920313202

[64] S. Marchal, J. François, R. State, and T. Engel, “Phishstorm: Detecting phishing with streaming analytics,” IEEE Transactions on Network and Service Management, vol. 11, pp. 458–471, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14293273

[65] E.-S. M. El-Alfy, “Detection of phishing websites based on probabilistic neural networks and k-medoids clustering,” Comput. J., vol. 60, pp. 1745–1759, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:11540367

[66] R. Mohammad, “Phishing websites features,” 03 2015

[67] A. Das, S. Baki, A. El Aassal, R. Verma, and A. Dunbar, “Sok: A comprehensive reexamination of phishing research from the security perspective,” IEEE Communications Surveys & Tutorials, vol. PP, pp. 1–1, 12 2019

[68] Verisign, “The domain name industry brief - volume 20 - issue 4,” Verisign, Tech. Rep., November 2023. [Online]. Available: https://dnib.com/media/downloads/reports/pdfs/2023/domain-name-report-Q32023.pdf

[69] R. Nokhbeh Zaeem and K. Barber, “A large publicly available corpus of website privacy policies based on dmoz,” in Conference on Data and Application Security and Privacy 2021, 04 2021, pp. 143–148.

[70] StationX, "*Top Phishing Statistics for 2024: Latest Figures and Trends,*" Accessed on: Mar. 30, 2024. [Online]. Available: https://www.stationx.net/phishing-statistics/#:~:text=1.,trillion%20phishing%20emails%20per%20year.

[71] Proofpoint, "2023 state of the phish," Proofpoint, Tech. Rep., 2023. [Online]. Available: https://www.proofpoint.com/sites/default/files/threat-reports/pfpt-us-tr-state-of-the-phish-2023.pdf

[72] D. Alperovitch, "*Revealed: Operation Shady RAT*," McAfee, 2011, [Online]. Available: https://icscsi.org/library/Documents/Cyber_Events/McAfee%20-%20Operation%20Shady%20RAT.pdf

[73] Jain, A.K., Gupta, B.B. A machine learning based approach for phishing detection using hyperlinks information. J Ambient Intell Human Comput 10, 2015–2028 (2019). https://doi.org/10.1007/s12652-018-0798-z

[74] M. Sameen, K. Han and S. O. Hwang, "PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System," in IEEE Access, vol. 8, pp. 83425-83443, 2020, https://doi.org/10.1109/ACCESS.2020.2991403

[75] N. Abdelhamid, A. Ayesh, F. Thabtah "Phishing detection based Associative Classification data mining." in Expert Systems with Applications. 41, pp.5948–5959. 10.1016/j.eswa.2014.03.019.

[76] Mohammad, R.M., Thabtah, F. & McCluskey, L. Predicting phishing websites based on self-structuring neural network. Neural Comput & Applic 25, 443–458 (2014). https://doi.org/10.1007/s00521-013-1490-z

[77] Bruen, Garth. (2015). Using WHOIS. 10.1002/9781118985786.ch2.

[78] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, ''A systematic literature review on phishing email detection using natural language processing techniques,'' IEEE Access, vol. 10, pp. 65703–65727, 2022.

[79] G. Sonowal and K. S. Kuppusamy, ''PhiDMA—A phishing detection model with multi-filter approach,'' J. King Saud Univ.-Comput. Inf. Sci., vol. 32, no. 1, pp. 99–112, Jan. 2020.

[80] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, ''Accurate and fast URL phishing detector: A convolutional neural network approach,'' Comput. Netw., vol. 178, Sep. 2020, Art. no. 107275.

[81] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani, ''Systematization of knowledge (SoK): A systematic review of software-based web phishing detection,'' IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2797–2819, 4th Quart., 2017.

[82] K. Rendall, A. Nisioti, and A. Mylonas, ''Towards a multi-layered phishing detection,'' Sensors, vol. 20, no. 16, p. 4540, Aug. 2020.

[83] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, ''CANTINA+: A featurerich machine learning framework for detecting phishing web sites,'' ACM Trans. Inf. Syst. Secur., vol. 14, no. 2, pp. 1–28, Sep. 2011.

[84] G. Vrbancic, I. Fister, and V. Podgorelec, ‘‘Datasets for phishing websites detection,’’ Data Brief, vol. 33, Dec. 2020, Art. no. 106438. [Online]. Available: https://data.mendeley.com/datasets/72ptz43s9v/1

List Of Publications

[IS1] J. Bohacik, I. Skula and M. Zabovsky, "Data Mining-Based Phishing Detection," in *15th Conference on Computer Science and Information Systems (FedCSIS)*, 2020, pp. 27-30, doi: 10.15439/2020F140.

[IS2] I. Skula, J. Bohacik and M. Zabovsky, "Use of Different Channels for User Awareness and Education Related to Fraud and Phishing in a Banking Institution," in *18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2020, pp. 606-612, doi: 10.1109/ICETA51985.2020.9379220.

[IS3] I. Skula, "Automated detection techniques of phishing," in *Mathematics in Science and Technologies (MIST)*, 2021, pp. 60-67.

[IS4] I. Skula, M. Kvet, "Domain Blacklist Efficacy for Phishing Web-page Detection Over an Extended Time Period," in *33rd Conference of Open Innovations Association (FRUCT)*, 2023, pp. 257-263, doi:10.23919/FRUCT58615.2023.10142999.

[IS5] I. Skula, M. Kvet, "URL and Domain Obfuscation Techniques - Prevalence and Trends Observed on Phishing Data," in *IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics*, 2024, pp. 283-290, doi:10.1109/SAMI60510.2024.10432841.

[IS6] I. Skula, M. Kvet, "Phishing Webpage Longevity," in *WorldCist2024*, 2024, doi:10.1007/978-3-031-60328-0_21

[IS7] I. Skula and M. Kvet, "A Framework for Preparing a Balanced and Comprehensive Phishing Dataset," in IEEE Access, doi:10.1109/ACCESS.2024.3387437.

List Of Figures

List Of Tables

Appendix 1 – URL-based Characteristics

Source	ID	Variable Name	Description
Scheme	1	**sch_len**	Length of the scheme section
Scheme	2	**sch_http_flg**	Whether the scheme is (http)
Scheme	3	**sch_https_flg**	Whether the scheme is (https)
Scheme	4	**sch_ftp_flg**	Whether (ftp) is present
Scheme	5	**sch_data_flg**	Whether (data) is present
Scheme	6	**sch_mailto_flg**	Whether (mailto) is present
Scheme	7	**sch_unkn_flg**	whether the scheme is not (http, https, mailto, ftp, data)
Scheme	8	**sch_secure_flg**	Whether scheme is one of (https, ftps, sftp or ldaps)
URL Full	9	**url_len**	Length of the URL section
URL Full	10	**url_dot_cnt**	Number of dots (.)
URL Full	11	**url_hyphen_cnt**	Number of hyphens (-)
URL Full	12	**url_dash_cnt**	Number of dashes (—)
URL Full	13	**url_underscore_cnt**	Number of underscores (_)
URL Full	14	**url_slash_cnt**	Number of slashes (/)
URL Full	15	**url_questionmark_cnt**	Number of question marks (?)
URL Full	16	**url_equal_cnt**	Number of equal signs (=)
URL Full	17	**url_and_cnt**	Number of and signs (&)
URL Full	18	**url_exclamation_cnt**	Number of exclamation signs (!)
URL Full	19	**url_space_cnt**	Number of spaces ( )
URL Full	20	**url_tilde_cnt**	Number of tildes (~)
URL Full	21	**url_comma_cnt**	Number of commas (,)
URL Full	22	**url_plus_cnt**	Number of plus signs (+)
URL Full	23	**url_asterisk_cnt**	Number of asterisk signs (*)
URL Full	24	**url_percent_cnt**	Number of percent signs (%)
URL Full	25	**url_hash_cnt**	Number of hash signs (#)
URL Full	26	**url_dollar_cnt**	Number of dollars ($)
URL Full	27	**url_at_cnt**	Number of at signs (@)
URL Full	28	**url_digit_cnt**	Number of digits
URL Full	29	**url_digit_pct**	Percentage of digits
URL Full	30	**url_upcase_cnt**	Number of upcased characters
URL Full	31	**url_upcase_pct**	Percentage of uppercase letters, to detect encoding anomalies
URL Full	32	**url_tld_pos**	Position of the top-level domain in the URL,
URL Full	33	**url_enc_flg**	Whether the URL uses URL encoding (%XX where XX is a hexadecimal value)
URL Full	34	**url_punycode_flg**	Flag that puny code is present
Authority	35	**aut_num_domain_cnt**	Count of purely numeric sub-domains
Authority	36	**aut_max_domain_len**	Length of largest sub-domain in the authority
Authority	37	**aut_avg_domain_len**	Mean/average sub-domain length
Authority	38	**aut_med_domain_len**	Median sub-domain length

Authority	39	**aut_hirisk_keyword_cnt**	Count of defined keywords within authority segment (secure login verify account bank signon)
Authority	40	**aut_hirisk_30brand_cnt**	Count of defined keywords derived from most commonly attacked brands (office365 outlook bet365 att whatsapp telegram netflix dhl usps gazprom steam facebook apple orange allegro tencent instagram yahoo gricole adobe bancolombia garena coinbase amazon microsoft wells google paypal irs linkedin)
Authority	41	**aut_len**	Length of the authority section
Authority	42	**aut_dot_cnt**	Number of dots (.)
Authority	43	**aut_hyphen_cnt**	Number of hyphens (-)
Authority	44	**aut_dash_cnt**	Number of dashes (—)
Authority	45	**aut_underscore_cnt**	Number of underscores (_)
Authority	46	**aut_questionmark_cnt**	Number of question marks (?)
Authority	47	**aut_equal_cnt**	Number of equal signs (=)
Authority	48	**aut_and_cnt**	Number of and signs (&)
Authority	49	**aut_exclamation_cnt**	Number of exclamation signs (!)
Authority	50	**aut_space_cnt**	Number of spaces ( )
Authority	51	**aut_tilde_cnt**	Number of tildes (~)
Authority	52	**aut_comma_cnt**	Number of commas (,)
Authority	53	**aut_plus_cnt**	Number of plus signs (+)
Authority	54	**aut_asterisk_cnt**	Number of asterisk signs (*)
Authority	55	**aut_percent_cnt**	Number of percent signs (%)
Authority	56	**aut_slash_cnt**	Number of slashes (/)
Authority	57	**aut_vowel_cnt**	Number of vowels (a,e,i,o,u)
Authority	58	**aut_vowel_pct**	% share of vowels (a,e,i,o,u) in the aut_len
Authority	59	**aut_vowely_cnt**	Number of vowels (a,e,i,o,u + y)
Authority	60	**aut_vowely_pct**	% share of vowels (a,e,i,o,u + y) in the aut_len
Authority	61	**aut_ipv4_flg**	Flag that ipv4 is present (sourced from dom_ipv4_flg)
Authority	62	**aut_hex_ip_flg**	Flag that IP is written using hex coding
Authority	63	**aut_hex_url_flg**	Flag that hex is used within authority segment
Authority	64	**aut_www_flg**	Flag whether www is present
Authority	65	**aut_at_flg**	At sign present (@)
Authority	66	**aut_at_cnt**	Number of at signs present (@)
Authority	67	**aut_port_flg**	Port sign is present (:)
Authority	68	**aut_portnum_flg**	Port number is present
Authority	69	**aut_nonstd_port_flg**	Flag if a non-standard port is used, uncommon ports might be used for malicious purposes
Authority	70	**aut_server_client_flg**	Flag if 'server' or 'client' is present in the domain name
Authority	71	**aut_punycode_flg**	Flag that puny code is present
Authority	72	**aut_dom_cnt**	Number of subdomains, as multiple subdomains can be suspicious
Authority	73	**aut_hirisk_tld_flg**	High risk top-level domain (present in top 10% phishing tlds, not present in top 10% legitimate)
Authority	74	**aut_tld_top10P_flg**	TLD present in the top 10 phishing TLDs by share

Authority	75	**aut_tld_type_num**	Type of the TLD (dom_tld_type: 1. country-code, 2.generic, 3.generic-restricted, 4.infrastructure, 5.sponsored, 6.test, 0.unknown)
Authority	76	**aut_tld_dot_end_flg**	Dot at the end of the domain
Authority	77	**aut_urlshort_flg**	Whether authority contains url shortener
Authority	78	**aut_blacklist_d3_flg**	D3 form of the domain of the URL is in our Blacklist
Authority	79	**aut_blacklist_d4_flg**	D4 form of the domain of the URL is in our Blacklist
Authority	80	**aut_blacklist_d5_flg**	D5 form of the domain of the URL is in our Blacklist
Authority	81	**aut_greylist_d3_flg**	D3 form of the domain of the URL is in our Greylist
Authority	82	**aut_greylist_d4_flg**	D4 form of the domain of the URL is in our Greylist
Authority	83	**aut_greylist_d5_flg**	D5 form of the domain of the URL is in our Greylist
Path	84	**pth_len**	Length of the path section
Path	85	**pth_dot_cnt**	Number of dots (.)
Path	86	**pth_hyphen_cnt**	Number of hyphens (-)
Path	87	**pth_dash_cnt**	Number of dashes (—)
Path	88	**pth_underscore_cnt**	Number of underscores (_)
Path	89	**pth_slash_cnt**	Number of slashes (/)
Path	90	**pth_questionmark_cnt**	Number of question marks (?)
Path	91	**pth_equal_cnt**	Number of equal signs (=)
Path	92	**pth_and_cnt**	Number of and signs (&)
Path	93	**pth_exclamation_cnt**	Number of exclamation signs (!)
Path	94	**pth_space_cnt**	Number of spaces ( )
Path	95	**pth_tilde_cnt**	Number of tildes (~)
Path	96	**pth_comma_cnt**	Number of commas (,)
Path	97	**pth_plus_cnt**	Number of plus signs (+)
Path	98	**pth_asterisk_cnt**	Number of asterisk signs (*)
Path	99	**pth_percent_cnt**	Number of percent signs (%)
Path	100	**pth_hash_cnt**	Number of hash signs (#)
Path	101	**pth_dollar_cnt**	Number of dollars ($)
Path	102	**pth_at_cnt**	Number of at signs present (@)
Path	103	**pth_dir_cnt**	Number of sub directories in the path of the URL
Path	104	**pth_base64_flg**	Detects base64 encoding
Path	105	**pth_hex_flg**	Detects hex encoding
Path	106	**pth_max_depth**	Maximum path depth
Filename	107	**fnm_len**	Length of the filename section
Filename	108	**fnm_dot_cnt**	Number of dots (.)
Filename	109	**fnm_hyphen_cnt**	Number of hyphens (-)
Filename	110	**fnm_dash_cnt**	Number of dashes (—)
Filename	111	**fnm_underscore_cnt**	Number of underscores (_)
Filename	112	**fnm_questionmark_cnt**	Number of question marks (?)
Filename	113	**fnm_equal_cnt**	Number of equal signs (=)
Filename	114	**fnm_and_cnt**	Number of and signs (&)
Filename	115	**fnm_exclamation_cnt**	Number of exclamation signs (!)
Filename	116	**fnm_space_cnt**	Number of spaces ( )
Filename	117	**fnm_tilde_cnt**	Number of tildes (~)

Filename	118	**fnm_comma_cnt**	Number of commas (,)
Filename	119	**fnm_plus_cnt**	Number of plus signs (+)
Filename	120	**fnm_asterisk_cnt**	Number of asterisk signs (*)
Filename	121	**fnm_percent_cnt**	Number of percent signs (%)
Filename	122	**fnm_hash_cnt**	Number of hash signs (#)
Filename	123	**fnm_dollar_cnt**	Number of dollars ($)
Filename	124	**fnm_at_cnt**	Number of at signs present (@)
Filename	125	**fnm_exe_flg**	Presence of an executable file extension (.exe, .bin, .scr, .vbs, .bat)
Filename	126	**fnm_pdf_flg**	Presence of pdf extension
Filename	127	**fnm_jpg_flg**	Presence of jpg or jpeg extension
Filename	128	**fnm_png_flg**	Presence of png extension
Filename	129	**fnm_doc_flg**	Presence of doc or docx extension
Filename	130	**fnm_base64_flg**	Detects base64 encoding
Query	131	**qry_len**	Length of the query section
Query	132	**qry_dot_cnt**	Number of dots (.)
Query	133	**qry_hyphen_cnt**	Number of hyphens (-)
Query	134	**qry_dash_cnt**	Number of dashes (—)
Query	135	**qry_underscore_cnt**	Number of underscores (_)
Query	136	**qry_slash_cnt**	Number of slashes (/)
Query	137	**qry_questionmark_cnt**	Number of question marks (?)
Query	138	**qry_equal_cnt**	Number of equal signs (=)
Query	139	**qry_and_cnt**	Number of and signs (&)
Query	140	**qry_exclamation_cnt**	Number of exclamation signs (!)
Query	141	**qry_space_cnt**	Number of spaces ( )
Query	142	**qry_tilde_cnt**	Number of tildes (~)
Query	143	**qry_comma_cnt**	Number of commas (,)
Query	144	**qry_plus_cnt**	Number of plus signs (+)
Query	145	**qry_asterisk_cnt**	Number of asterisk signs (*)
Query	146	**qry_percent_cnt**	Number of percent signs (%)
Query	147	**qry_hash_cnt**	Number of hash signs (#)
Query	148	**qry_dollar_cnt**	Number of dollars ($)
Query	149	**qry_at_cnt**	Number of at signs present (@)
Query	150	**qry_params_cnt**	Number of parameters in the query
Fragment	151	**fgm_len**	Length of the fragment
Fragment	152	**fgm_token_cnt**	Count of distinct tokens in the fragment
Fragment	153	**fgm_specialchar_cnt**	Count of special characters, which can be used in XSS or other attacks
Fragment	154	**fgm_hashbang_flg**	Flag if the fragment starts with "!#", used in Ajax-heavy sites but can be abused
Fragment	155	**fgm_script_presence_flg**	Flag for the presence of script-like or executable content in the fragment

Appendix 2 – HTML-based Characteristics

Source	ID	Variable Name	Description
HTML	1	**htm_len**	Length of the file
HTML	2	**htm_whitespace_cnt**	Number of whitespace
HTML	3	**htm_nowhtspace_len**	Length of the file without whitespaces
HTML	4	**htm_visible_len**	Length (number of characters) of the text visible in the screen
HTML	5	**htm_invisible_len**	Length (number of characters) of invisible text (tags, objects, etc.)
HTML	6	**htm_vis_ratio_pct**	Share of visible vs. invisible characters
HTML	7	**htm_invis_to_vis_pct**	Ratio of HTML code to visible text, indicating the amount of markup relative to content, high in deceptive pages
HTML	8	**htm_script_tag_pct**	Proportion of <script> tags relative to the total number of tags
HTML	9	**htm_input_tag_pct**	Proportion of <input> tags relative to the total number of tags
HTML	10	**htm_comment_density**	Ratio of the number of characters in comments to the total HTML length
HTML	11	**htm_inv_elem_cnt**	Count of elements styled with display:none or visibility:hidden
HTML	12	**htm_freq_meta_ref_cnt**	Count of <meta> tags with a refresh directive triggering in less than 5 seconds
HTML	13	**htm_obf_script_cnt**	Count of <script> tags containing patterns commonly used in obfuscation (e.g., eval, window.atob)
HTML	14	**htm_susp_attr_cnt**	Count of tags with attributes like onmouseover, onclick containing JavaScript code
HTML	15	**htm_data_uri_cnt**	Count of tags using src="..." or similar
HTML	16	**htm_inline_stl_cnt**	Count of tags with inline styles
HTML	17	**htm_comment_len**	Length (number of characters) of comments within the HTML file
HTML	18	**htm_total_tag_length**	The total length of all HTML tags in a document
HTML	19	**htm_tag_density**	Ratio of tag characters occupied by tags divided by total HTML length
HTML	20	**htm_script_cont_len**	Length of content within <script> tags
HTML	21	**htm_style_cont_len**	Length of content within <style> tags or 'style' attributes
HTML	22	**htm_form_act_ext_cnt**	Count of forms whose action attributes point to external domains
HTML	23	**htm_input_all_types_cnt**	Overall count of different input types (text, password, submit, etc.)
HTML	24	**htm_input_file_cnt**	Count of input objects for file manipulation
HTML	25	**htm_input_checkbox_cnt**	Count of input checkbox objects

HTML	26	**htm_input_text_cnt**	Count of input objects for text
HTML	27	**htm_input_password_cnt**	Count of input objects for passwords
HTML	28	**htm_input_submit_cnt**	Count of input submit objects
HTML	29	**htm_lrg_image_cnt**	Count of <img> tags with large file sizes specified in attributes or by inference
HTML	30	**htm_audio_video_cnt**	Count of multimedia elements such as <audio> and <video> tags
HTML	31	**htm_autoexec_scp_cnt**	Count of script tags or inline scripts that execute automatically without user interaction
HTML	32	**htm_form_get_cnt**	Count of <form> tags using the GET method
HTML	33	**htm_meta_keywords_cnt**	Count of <meta> tags with a name attribute of "keywords"
HTML	34	**htm_trademark_cnt**	Number of occurrences of trademark special character
HTML	35	**htm_copyright_cnt**	Number of occurrences of copyright special character
HTML	36	**htm_registered_cnt**	Number of occurrences of registered special character
HTML	37	**htm_ext_script_cnt**	Count of <script> tags sourcing from external domains
HTML	38	**htm_int_script_cnt**	Count of <script> tags sourcing from the same domain
HTML	39	**htm_script_safe_src_pct**	Ratio of scripts loaded from secure (HTTPS) sources to all script tags
HTML	40	**htm_link_obf_flg**	Flag if there are obfuscated links, such as those using URL shorteners
HTML	41	**htm_href_ext_cnt**	Count of links pointing outside the domain
HTML	42	**htm_href_int_cnt**	Count of links pointing inside the domain
HTML	43	**htm_href_ext_pct**	Ratio of links to other domains to all links
HTML	44	**htm_href_susp_flg**	Links on the page contains @ or -
HTML	45	**htm_href_ext_obj_pct**	Ratio of videos/images/scripts from other domains to all links
HTML	46	**htm_href_blank_target_cnt**	Count of links with target="_blank" without rel="noopener noreferrer"
HTML	47	**htm_href_javascript_cnt**	Count of links that use "javascript:" in the href attribute
HTML	48	**htm_no_follow_links_cnt**	Count of links with rel="nofollow"
HTML	49	**htm_tag_input_cnt**	Flag that form <input> is present
HTML	50	**htm_tag_form_cnt**	Flag that <form> is present
HTML	51	**htm_tag_script_cnt**	Count of <script> tags
HTML	52	**htm_tag_a_cnt**	Count of <a> (anchor) tags
HTML	53	**htm_tag_external_a_cnt**	Count of <a> tags with URLs pointing to different domains than the base URL
HTML	54	**htm_tag_js_a_cnt**	Count of <a> tags using "javascript:" in the href attribute
HTML	55	**htm_tag_img_cnt**	Count of <img> (image) tags
HTML	56	**htm_tag_img_no_alt_cnt**	Count of images without alt tags

HTML	57	**htm_tag_img_external_cnt**	Count of <img> tags with external sources
HTML	58	**htm_tag_img_internal_cnt**	Count of <img> tags placed within domain
HTML	59	**htm_tag_iframe_cnt**	Count of <iframe> tags
HTML	60	**htm_tag_meta_ref_cnt**	Count of <meta> tags with a refresh directive
HTML	61	**htm_tag_a_secure_link_cnt**	Count of HREF links using HTTPS
HTML	62	**htm_tag_a_insecure_link_cnt**	Count of HREF links using HTTP
HTML	63	**htm_tag_input_hidden_cnt**	Count of hidden <input> tags often used in deceptive practices
HTML	64	**htm_tag_hidden_input_value_cnt**	Count of <input> tags with type "hidden" that contain a non-empty value attribute
HTML	65	**htm_tag_button_cnt**	Count of <button> tags
HTML	66	**htm_tag_script_inline_cnt**	Count of <script> tags containing inline JavaScript
HTML	67	**htm_tag_style_inline_cnt**	Count of <style> tags or inline styles
HTML	68	**htm_tag_applet_cnt**	Count of <applet> tags, older but still potentially harmful
HTML	69	**htm_tag_embed_cnt**	Count of <embed> tags
HTML	70	**htm_tag_object_cnt**	Count of <object> tags
HTML	71	**htm_wrd_trademark_cnt**	Number of occurrences of the trademark word
HTML	72	**htm_wrd_copyright_cnt**	Number of occurrences of the copyright word
HTML	73	**htm_wrd_registered_cnt**	Number of occurrences of the registered word
HTML	74	**htm_wrd_login_cnt**	Count of the word "login" in the text
HTML	75	**htm_wrd_register_cnt**	Count of the word "register" in the text
HTML	76	**htm_wrd_password_cnt**	Count of the word "password" in the text
HTML	77	**htm_wrd_verify_cnt**	Count of the word "verify" in the text
HTML	78	**htm_wrd_bank_cnt**	Count of the word "bank" in the text
HTML	79	**htm_wrd_account_cnt**	Count of the word "account" in the text
HTML	80	**htm_wrd_update_cnt**	Count of the word "update" in the text
HTML	81	**htm_wrd_free_cnt**	Count of the word "free" in the text
HTML	82	**htm_wrd_prize_cnt**	Count of the word "prize" in the text
HTML	83	**htm_wrd_click_here_cnt**	Count of the phrase "click here" in the text
HTML	84	**htm_wrd_secure_lst_cnt**	Count of the word "secure", "safety" or "protected" in the text

Appendix 3 – 3rd Party-based Characteristics

Source	ID	Variable Name	Description
WHOIS	1	**who_reg_iana_id**	IANA ID for the registrar
WHOIS	2	**who_hirisk_iana_flg**	Flag for high-risk IANA ID
WHOIS	3	**who_IANA_top10P_flg**	Flag for presence in top 10 phishing IANAs
WHOIS	4	**who_days_age**	Domain age in days since creation
WHOIS	5	**who_reg_less_1Y_flg**	Flag for domain registration less than a year old
WHOIS	6	**who_days_to_expiry**	Number of days till domain expiry
WHOIS	7	**who_expiry_soon_flg**	Flag for domains expiring soon (less than 1 month)
WHOIS	8	**who_days_from_refresh**	Number of days since last domain update
WHOIS	9	**who_recent_update_flg**	Flag for recent domain update (within the last month)

Source	ID	Variable Name	Description
PING	1	**pin_url_match**	Flag if `url_out` matches `url_back`
PING	2	**pin_live_status**	Status of the URL being live or dead

Source	ID	Variable Name	Description
IPGEO	1	**ipg_cntry_iso_num**	ISO numeric code for the country
IPGEO	2	**ipg_hirisk_cntry_flg**	Flag if the IP belongs to a high-risk country
IPGEO	3	**ipg_cntry_top10P_flg**	Flag for country presence in top 10 phishing countries by prevalence

Appendix 4 – Sample Raw Whois Response For uniza.sk

```
Domain:                         uniza.sk
Created:                        2004-11-26
Valid Until:                    2024-11-26
Updated:                        2023-12-07
Domain Status:                  ok
Nameserver:                     nic.uniza.sk
Nameserver:                     proxy.uniza.sk
Nameserver:                     sun.uakom.sk

Domain registrant:              IUNI-0002
Name:                           Žilinská univerzita v Žiline
Organization:                   Žilinská univerzita v Žiline
Organization ID:                397563
Phone:                          +421.415131851
Email:                          lubos.kojdjak@uniza.sk
Street:                         Univerzitná 8215/1
City:                           Žilina
Postal Code:                    01026
Country Code:                   SK
Created:                        2017-09-01
Updated:                        2024-04-16

Registrar:                      IUNI-0002
Name:                           Žilinská univerzita v Žiline
Organization:                   Žilinská univerzita v Žiline
Organization ID:                397563
Phone:                          +421.415131851
Email:                          lubos.kojdjak@uniza.sk
Street:                         Univerzitná 8215/1
City:                           Žilina
Postal Code:                    01026
Country Code:                   SK
Created:                        2017-09-01
Updated:                        2024-04-16

Administrative Contact:         IUNI-0002
Name:                           Žilinská univerzita v Žiline
Organization:                   Žilinská univerzita v Žiline
Organization ID:                397563
Phone:                          +421.415131851
Email:                          lubos.kojdjak@uniza.sk
Street:                         Univerzitná 8215/1
City:                           Žilina
Postal Code:                    01026
Country Code:                   SK
Created:                        2017-09-01
```

Updated:	2024-04-16
Technical Contact:	IUNI-0002
Name:	Žilinská univerzita v Žiline
Organization:	Žilinská univerzita v Žiline
Organization ID:	397563
Phone:	+421.415131851
Email:	lubos.kojdjak@uniza.sk
Street:	Univerzitná 8215/1
City:	Žilina
Postal Code:	01026
Country Code:	SK
Created:	2017-09-01
Updated:	2024-04-16

Appendix 5 – Best Performing Features Comparison Table

	Top5	Top10	Top15	Top20	Top25a	Top25b	Top30
aut_avg_domain_len			✓	✓	✓	✓	✓
aut_blacklist_d3_flg	✓	✓	✓	✓	✓	✓	✓
aut_blacklist_d4_flg			✓	✓	✓		
aut_blacklist_d5_flg			✓	✓	✓	✓	✓
aut_dom_cnt				✓		✓	✓
aut_dot_cnt					✓	✓	✓
aut_greylist_d3_flg			✓	✓	✓		✓
aut_hirisk_30brand_cnt		✓	✓	✓	✓	✓	✓
aut_hirisk_keyword_cnt					✓	✓	✓
aut_hirisk_tld_flg		✓	✓	✓	✓		✓
aut_len						✓	✓
aut_tld_top10p_flg		✓	✓	✓	✓		✓
aut_urlshort_flg		✓	✓	✓	✓		
aut_www_flg	✓	✓	✓	✓	✓	✓	✓
fgm_len						✓	✓
fgm_token_cnt						✓	✓
fnm_hyphen_cnt					✓		
fnm_len						✓	✓
fnm_pdf_flg		✓		✓			
htm_audio_video_cnt					✓		
htm_form_get_cnt					✓		
htm_href_susp_flg	✓	✓	✓	✓	✓	✓	✓
htm_input_password_cnt					✓		
htm_link_obf_flg							✓
htm_wrd_trademark_cnt					✓		
pth_dir_cnt						✓	
pth_dot_cnt						✓	✓
pth_equal_cnt							✓
pth_len						✓	✓
pth_space_cnt						✓	
qry_asterisk_cnt							✓
qry_len						✓	✓
sch_http_flg				✓	✓	✓	✓
sch_secure_flg						✓	
url_hash_cnt							✓
url_len						✓	✓
url_questionmark_cnt						✓	
url_space_cnt			✓	✓	✓	✓	✓
url_upcase_pct			✓	✓	✓	✓	✓
who_expiry_soon_flg	✓	✓	✓	✓	✓	✓	✓
who_hirisk_iana_flg	✓	✓	✓	✓	✓		✓
who_recent_update_flg				✓	✓		
who_reg_less_1y_flg				✓	✓		✓

✓ - indicates an entirely new feature that was uniquely used in Top30 only