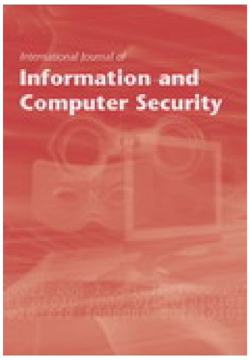


International Journal of Information and Computer Security

ISSN online: 1744-1773 - ISSN print: 1744-1765
https://www.inderscience.com/ijics

AI-driven approach for robust real-time detection of zero-day phishing websites

Thomas Nagunwa

DOI: 10.1504/IJICS.2024.10061314

Article History:

Received:	23 March 2023
Last revised:	03 September 2023
Accepted:	08 September 2023
Published online:	19 February 2024

AI-driven approach for robust real-time detection of zero-day phishing websites

Thomas Nagunwa

Department of Computer Science,
Institute of Finance Management,
Dar Es Salaam, Tanzania
Email: tom.nag@gmail.com

Abstract: Existing solutions for detecting phishing websites mainly depend on a blacklist approach, which has proven ineffective in detecting zero-day phishing websites in real-time. This study proposes a machine learning (ML) approach for highly accurate real-time detection of zero-day phishing websites using highly diversified features. The prediction performance of the features is evaluated and compared using 12 traditional ML and three deep learning (DL) algorithms. The results have shown that with CAT boost algorithm, the features are able to achieve the best performance with an accuracy of 99.02%, false positive rate (FPR) of 0.90% and false negative rate (FNR) of 1.03%. Feature analysis used to understand the features' prediction importance, data distributions and performance contributions are also presented. The prediction runtime of the proposed model is also measured to assess whether the model can be deployed for real-time detection.

Keywords: phishing; phishing web page; zero-day phishing web page; machine learning; deep learning.

Reference to this paper should be made as follows: Nagunwa, T. (2024) 'AI-driven approach for robust real-time detection of zero-day phishing websites', *Int. J. Information and Computer Security*, Vol. 23, No. 1, pp.79–118.

Biographical notes: Thomas Nagunwa has worked with the Institute of Finance Management for over ten years as a Lecturer. He received his Bachelor's degree from the University of Dar Es Salaam, Tanzania, Master's degree from Dublin Institute of Technology (Ireland) and PhD degree from Birmingham City University, UK. His research interests are in applied machine learning in various high impactful domains and cybersecurity in the internet.

1 Introduction

The rapid growth of global internet usage in recent years has led to a boom in online services in domains such as e-commerce, social networking and e-government. This has resulted in a surge in the volume of transactions of sensitive information such as personal data on the internet. Online availability of such data has lured attackers to devise a cyberattack mechanism, known as phishing, to enable them to steal the data and use it to impersonate victims for undertaking malicious activities. Today, phishing is one of the most important and effective types of cyberattacks and has caused massive economic, political and social impacts on individuals, organisations and governments (Alkhalil et al., 2021; Allianz, n.d.; Ball, 2017; Brattberg and Maurer, 2018; CNN, 2020; FBI, 2018; Gendre, 2015, 2019; Greenberg, 2017; IBM Security, 2019; Internet Society, 2016; Koulopoulos, 2017; Lee and Rotoloni, 2016; Pompon, 2019; Ponemon Institute, 2015; Retruster, n.d.; Rodríguez, 2019; SecureWorks, 2019; Sophos, 2019; Verizon, 2018).

The execution of a successful phishing attack involves two main stages namely the distribution/delivery of the attack to the targeted users and the capturing of victims' data. Attackers use various techniques such as phishing spams to distribute the attacks to their targets. In the second stage, the use of phishing websites remains the main tactic used by attackers to capture data (PhishLabs, 2020). A phishing website is a replica of a legitimate website that prompts users for similar personal data to that requested by the legitimate website. Today, with the help of highly sophisticated and automated phishing toolkits, which are widely available at a low cost, high-quality phishing websites are being developed even by technically unskilled attackers. This has resulted in a rise in the number of newly undiscovered (zero-day) phishing websites created on a daily basis (Damiani, 2020; Sophos, 2019).

There are a number of developed solutions for detecting phishing websites at the time users are exposed to access them. However, they mainly rely on a blacklist approach, which has been demonstrated to be inefficient in detecting zero-day phishing websites in real-time (Barraclough et al., 2013; Wenyin et al., 2012). Researchers have proposed various approaches including those based on visual similarity and rules. The former, however, offer limited protection, that is, they only detect phishing websites whose legitimate web pages are recorded in the database. Meanwhile, most rule-based solutions are limited with low or moderate performances and/or low diversity of features used, increasing their susceptibility to detection evasions. Due to these limitations, the number of zero-day phishing websites and their successful attacks has been steadily growing globally over the years. For instance, APWG (2023) reported that the number of phishing attacks has grown by more than 150% per year between 2019 and 2022. They also observed that the number of new unique phishing websites per month reached the highest recorded number of 459,139 in December 2022, a sharp rise from 69,533 recorded by APWG (2016) in the same month in 2016. On the other hand, the Federal Bureau of Investigations (FBI, 2021) reported that the number of complaints they received from phishing victims rose from 25,344 in 2017 to 323,972 in 2021.

Given the prevalence and significance of zero-day phishing websites, an ideal solution for effective and efficient detection of the websites should have the following design characteristics:

- It must not rely on lists of known or suspected phishing websites compiled from human or software-generated reports.
- It preferably uses machine learning (ML) due to its flexibility in updating the prediction rules through data re-collection and re-training. This is useful in maintaining the optimal detection performance when phishing website techniques change over time.
- It must achieve high prediction accuracy and low misclassification rates, ideally, 100% and 0% respectively.

- It must perform detection in real-time, i.e., the additional time taken to determine whether a URL will take the user to a phishing web page or not must not degrade the user's overall web browsing experience.
- It should use novel prediction features. This is because the attackers tend to learn prediction features used by existing detection solutions and develop mechanisms to circumvent the solutions. The use of novel features will enable the solution to be ahead of attackers.
- It should use highly diversified prediction features (features selected from a wide variety of categories) to make the solutions more resistant to detection evasion. The attackers would need to develop at least one detection evasion technique for each feature category to have any chance of eluding the solution, which is likely to be a difficult and time-consuming task for most attackers.

This study, therefore, proposes and evaluates a new set of largely diversified features for highly accurate real-time prediction of zero-day phishing web pages using an ML approach. To identify the best ML algorithm for the task, various traditional and deep learning (DL) ML algorithms are used to evaluate the features and their results are compared using several standard ML performance metrics to identify the best performing algorithm.

This paper is arranged into seven sections. Section 2 describes works related to this study. In Section 3, a background to structural characteristics of phishing web pages is provided. Section 4 describes the proposed prediction features and the design of our prediction model. In Section 5, we describe the experiments for developing and evaluating the model and present their results along with feature analysis. Section 6 compares our work with other related works and discusses the applicability and limitations of our solution. Section 7 concludes the paper by revisiting our results and contributions.

2 Related work

Current approaches for detecting phishing web pages can be grouped into three main categories. These are blacklists, visual similarity and rule-based techniques.

2.1 Blacklist-based approach

In this category, several web browser filters and anti-malware suites are available to protect users from accessing phishing web pages. The filters are incorporated in web browsers as a built-in or installable component (also known as a plug-in). The anti-malware suites, on the other hand, are software that can either be installed in a user's machines as standalone applications or as clients of cloud-based applications. The suites scan the websites as they are being accessed by users to detect malicious behaviours. Most filters and suites are based on a URL blacklist approach. Google's Safe Browsing, for instance, uses a database of hashed URLs of malicious web pages, including phishing ones, to protect users of Chrome, Firefox, Safari and Opera web browsers. It warns users when they access phishing web pages whose URLs are in the list (Google, n.d.-a, n.d.-b; Somogyi and Miller, 2017; Geotrust, n.d.). Examples of plug-ins using blacklists are

TrustWatch, Bitdefender TrafficLight and PhishTank SiteChecker (EC Council, 2017; Geotrust, n.d.; Kent, 2013) whereas anti-malware suites include Trend Micro (n.d.), ESET (2017) and Kaspersky (2015).

The blacklist approach, however, is less effective in instantly detecting zero-day phishing webpages. This is because blacklists mainly depend on users or experts reporting phishing webpages before their administrators verifying them prior to be recorded in the database, the process which takes time leading to delays in updating the lists. By the time the lists are updated, the reported phishing websites are likely to have been active for several hours, days or weeks. The ineffectiveness was empirically demonstrated by various studies including Wenyin et al. (2012), Barraclough et al. (2013) and AV-Comparatives (2016).

2.2 Visual similarity approach

This approach compares images of suspicious web pages or in combination with web page structure and contents often against those of pre-collected legitimate web pages to detect phishing web pages based on the computed visual similarity scores. Hara et al. (2009) proposed a technique in which a domain name and image of a web page (screenshot) are compared against a database of domain names and images of pre-collected legitimate and phishing web pages. A new web page is flagged as a phishing web page if its visual similarity score against one of the web pages in the database is higher than the pre-calculated threshold and its domain name is not matching with any of those in the database. Phishing web pages were used in the database to help detecting other phishing web pages targeting the same legitimate web page as they are likely to be similar in their visual looks. The technique achieved an accuracy of 80% and FPR of 17.5%. Other works based on visual based features approach include Medvet et al. (2008), Chen et al. (2009) and Kumar and Kumar (2015). The disadvantage of this approach, as observed by Medvet et al. (2008), is that the extraction of visual related features is computationally expensive, that is, it consumes more computational resources and introduces significant detection overheads which may not be suitable for real-time detection.

2.3 Rule-based approach

This is one of the most popular approaches in research works because of its capability to instantly predict both known and unknown phishing web pages with good performances. This category uses rules that are set manually or determined automatically by ML algorithms to distinguish phishing web pages from legitimate ones. Xiang et al. (2011), for instance, developed a CANTINA+ system, based on a Bayesian network classifier, to detect phishing web pages. The classifier used 15 features related to web page contents, WHOIS domain records, URL structure and search engine reputation to detect phishing web pages to achieve accuracy, FPR and F1 scores of 92.25%, 1.375% and 0.95 respectively. Shirazi et al.'s (2017) work compared several DL algorithms and SVMs to detect phishing web pages. Using 30 features related to URL structure, web page structure, WHOIS domain records, Alexa's web page reputation and Google's search engine reputation, one of the DL algorithms achieved the best results with an area under curve (AUC) ROC of 0.897, true negative rate (TNR) of 90.27%, and true positive rate (TPR) of 89.33%. Jain and Gupta's (2018) random forest classifier detected phishing web pages with an accuracy of 99.09% and an FPR of 1.25%. A total of 19 features based on URL structure, and web page structure and contents were used by the classifier. A Random Forest classifier proposed by Sahingoz et al. (2019) used vector representations of URL characters as well as the website's ranking in the Alexa top websites list to detect phishing web pages. An accuracy, precision, sensitivity and f-measure of 97.97%, 0.97, 0.99 and 0.98 respectively were achieved by the classifier. Li et al. (2019) proposed a classifier composed of a stack of boosting algorithms (XGBoost and LightGBM) to detect phishing web pages. The classifiers used 20 features based on URL structure, and web page structure and contents to yield an accuracy of 97.3%, FPR of 4.46% and FNR of 1.61%.

A study by Elsadig et al. (2022) developed a deep convolutional neural network (CNN) to distinguish phishing URLs from legitimate ones from a dataset of 549,346 URLs. First, they extracted a set of 12 most significant features from the URLs' characters using a bidirectional encoder representations from transformers (BERT) pre-trained model and used the features to train the CNN model. The model achieved the optimal accuracy of 96.66%, precision of 96.66% and f1 score of 93.63%. Liu et al. (2022) proposed a DL-based phishing detection model that utilised semantic analysis of various text-based components of a webpage including URL, title, and HTML body and invisible texts. The extracted texts of each component, from a dataset of 5,393 web pages, were independently converted into vector representations and passed through multiple convolution layers of each component's CNN model for feature extraction. Then their outputs were fused together in various combinations for the classification task by a single fully connected neural network layer. The model's output was able to achieve the best performance of FPR of 0.0047, F1 score of 0.9830 and AUC of 0.9993 with a combination of URL, title, and body and invisible text. Alshingiti et al.'s (2023) work evaluated a set of 80 URL character-based features of a dataset of 20,000 phishing and benign URLs using three DL algorithms namely CNN, LSTM and LSTM-CNN to detect phishing URLs. First, they applied a SelectKBest feature selection method to identify the most predictive features (30) and compared the performances of the features with the three algorithms. The results showed that CNN outperformed the other algorithms by obtaining the best accuracy of 99.20%.

Despite being the most successful approach compared to others, the proposed solutions have several limitations. One, some of these, for instance, Xiang et al. (2011), Shirazi et al. (2017) and Elsadig et al. (2022), have achieved moderate prediction performances as indicated earlier. Two, most of these approaches used sets of features with low diversity. Examples of this are Jain and Gupta (2018), Sahingoz et al. (2019) and Li et al. (2019) which used only two different categories of features (URL structure, and web page structure and contents) while Elsadig et al. (2022) and Alshingiti et al. (2023) used only one (URL structure). With a small set of feature categories, attackers can learn most or all the features belonging to each category and require one or a few evasion mechanisms to bypass each category, thus the entire solution. Three, some of these solutions avoided using featured based on third party services for performance reasons. Instead, they used only features derived from URL and/or web page structures. The later, however, can easily be emulated by phishers by ensuring that the phishing webpages and their URLs are as similar as their targets, thus neutralising the features' prediction power. We argue that it is extremely difficult for phishers to manipulate third party services because they are highly secured. To manipulate the services, phishers would require high technical skills, longer time and other resources that very few attackers are likely to possess or willing to invest in.

3 Structural characteristics of data capturing web pages

This section describes the structure of web pages that collect personal data from users in order to provide a background of our proposed features.

3.1 Anatomy of a web page collecting personal data

A web page is a hypertext document, usually in a hypertext markup language (HTML) format, that is viewed remotely through a web browser. We term a web page that prompts for and collects personal data a *personal data capturing (PDC) web page*. Each web page has two main parts: uniform resource locator (URL) and the HTML file describing its structure and contents.

3.1.1 URL

URL, also referred to as a web page address, is a unique string identifying a location of a web page file on the internet. A URL has three main components namely the protocol, hostname and path. For instance, for a URL *https://cs.berkeley.edu/resources/faculty-staff*, *https* is a protocol, *cs.berkeley.edu* is the name of the host (hostname) and resources/faculty-staff is a path. Hostnames have a hierarchical structure: *edu* is a top-level domain used by educational organisations, *berkeley* is a sub-domain of *edu* owned by the University of Berkeley, and *cs* is a sub-domain of *berkeley.edu* issued to the Berkeley Computer Science Department. The hostname denotes a website (host), and the path gives the location of the web page in the file system of the website. To download a web page from its server to a client machine (e.g., in order to display it in a browser), a communication protocol known as hypertext transfer protocol (HTTP) or its encrypted version HTTP secure (HTTPS) are normally used. These two are indicated as *http* and *https* in the URL respectively. To use HTTPS, the owner of the website must acquire a transport layer security (TLS – formerly SSL) certificate for the domain. There are three main types of TLS certificates offered namely extended validation (EV), organisation validation (OV) and domain validation (DV) (Acmetek, n.d.; Global Sign, n.d.; Kavya, 2020; Robertckl, 2014; Warburton and Pompon, 2019). Not only is the certificate useful for encrypting the traffic but it is also used to authenticate hosts of websites.

3.1.2 Web page structure and contents

The components of an HTML web page structure are built around a basic element known as a tag. Examples of tags are <*title*>…. </*title*>, <*meta*….>, <*link*….> and <*script*>…. </*scripts*>. The HTML web page structure is made up of two sections; head and body. The head section may contain components including title, metadata, links and scripts. Links provide addresses of other web pages or objects, such as images and stylesheets, connected to the web page. There are three types of links namely links to other web pages' objects, links to various sections of the same web page and void links (those with no URL assigned) (Ghobril, 2015; Kurtus, 2014; Lakshmi and Vijaya, 2012; Omg, 2009).

Some web pages are created in multiple versions for various reasons such as identifying one consistent web page for search engine results and translation of a web page in multiple languages. To identify the related versions, canonical and alternate links are included in the head section with URLs of all the related web pages (Eubanks, 2012; Google, 2017; Meier, 2014; Microformats, 2016; Valk, 2016).

The main component of the body section of the PDC web page is the mechanism for capturing and submitting user data. The web page can use an HTML form or a script-based dialogue window for the task. The HTML form is a structural component of a web page for collecting input data from the user and sending it to a specified URL for processing. The form is delimited by a tag <*form*>… </*form*> and often contains various input fields, each defined with an input tag <*input*> and an attribute type for specifying the type of data to be collected. Common input types are *text*, which is a field accepting any text information, and *password*, which is a field specifically for password entries (W3Schools, n.d.-b). Others include email for email addresses, tel for telephone numbers, and date for day, month and year entries (W3Schools, n.d.-b). The form uses a URL assigned to its tag action to identify the URL to send the collected data to for processing tasks, such as saving the data into the database. The data processing web page is also known as the form handler. <*form action*="*/login*" *method*="*post*"> is an example of a form tag indicating a form handler named login within the same host as the PDC web page.

Script based dialogue windows which also can be used to prompt for personal data are often designed using JavaScript or JQuery scripting languages. With JavaScript, the *window.prompt()* command displays a prompting message and captures the inputs (Universal Class, n.d.-b; W3Schools, n.d.-c). Alternatively, both JavaScript and JQuery can incorporate the HTML form to prompt inputs using input fields (Agarwal, n.d.; Universal Class, n.d.-a).

Not every web page with an HTML form or a dialogue window collects personal data. For instance, Google's search web page collects users' search keywords. What differentiates PDC web pages from others is that they usually contain words or phrases that are related to the specific data being collected. These phrases, we refer to them as *PDC phrases*, can be within the form text (W3Schools, n.d.-b), in the label tags <*label*>….</*label*> (W3Schools, n.d.-a), as default values to value attribute of input tags (Broadley, n.d.) or elsewhere in the HTML document. Some of these phrases, for instance, login, log in and sign in can also be used as names of a submit button of the form. In this case, they are assigned as values to the input type submit.

3.1.3 Structural characteristics of phishing websites

Phishing websites tend to imitate the URL, layout and contents of legitimate websites as much as possible to lure users and evade detection. The degree of similarity between phishing and legitimate websites varies depending on the phisher's skills in reproducing the replica. Phishing websites can be categorised into three types depending on their levels of look and feel relative to the legitimate websites they imitate (Zhao et al., 2016). These are simple phishing, advanced phishing and extreme phishing websites. Three main criteria to describe the categories are summarised in Table 1. Using phishing toolkits, which often consist of loaded templates of the targeted legitimate websites, phishers can easily produce phishing websites that look closely like legitimate ones by performing a few modifications on the structure and contents of the templates. For instance, they have to change the form handler to point to the URL of a host that they will use to collect the phished data. Given the popularity of the toolkits among phishers, advanced and extreme phishing websites are likely to be the majority of the phishing websites created today

Table 1 Types of phishing websites are categorised based on the three-similarity metrics

Type of phishing websites	*Similarity metrics*		
	Visual appearance	*Page depth*	*Supports user dynamic interaction*
Simple phishing	Somewhat similar	One web page with few similar links	No
Advanced phishing	Mostly similar	Limited number of pages with few similar links	No
Extreme phishing	Similar in every way	Unlimited number of pages with completely similar links	Yes

Source: Zhao et al. (2016)

It is however impossible for a phishing web page to use the same URL as that of a legitimate web page. This is because the website's domain name and URL are the unique registered information for every website in the internet space. To imitate the original URLs or to hide the true identity of their suspicious URLs, phishers use two approaches. The first one is to compromise a web server hosting a legitimate website and add a phishing web page or website in the folder containing the legitimate website. The phishing web page/website, in this case, will be running as web page(s) of the legitimate website, thus, using the same domain name registered with the legitimate website but with different URL path(s) assigned by the phisher.

The second approach is to use their registered domain names but make them look like the legitimate ones or mask them using other unsuspicious characters. Various techniques are used to achieve this. These include the use of legitimate domain names in non-standard positions in their URLs (Xiang et al., 2011), replacing domain names with numerical digits or IP addresses (Xiang et al., 2011), encoding the domains names with other string presentation formats (e.g., ASCII characters) (PCHelp, 2002), the addition of various obfuscation characters (e.g., '-', '_', '=') (Ma et al., 2009), replacing their original URLs with shortened ones (Webroot, 2019) and the use of free domain names provided by free web hosting services (PhishLabs, 2020). These practices are not commonly associated with legitimate URLs, especially for those of PDC web pages. Due to the addition of more characters by some of these techniques, the resulting phishing URLs tend to be longer in length than those of legitimate websites. Also, in most legitimate websites, the domain names often represent brand names or names of the organisations owning the websites. These names usually appear multiple times in the contents or structural components of the web pages. Since in this approach the phishers' own domain names are different from the legitimate ones, they are less likely to relate to the contents or structural components of the phishing web pages, which are often copies of those of legitimate web pages.

Based on the structural designs of the phishing websites described above, several key structural differences between phishing web pages/websites and those of legitimate ones can be noticed. We summarise them in Table 2. These can be useful in determining potential features for differentiating the two, thus predicting phishing PDC web pages.

Table 2 A summary of key structural differences between legitimate and phishing web pages

Comparison factors	*Legitimate web pages/websites*	*Phishing web pages/websites*
Originality of web page structure and contents	Use their original structure and contents	Copy most of the structural components and contents of the target legitimate web pages including web page links. Few links may be modified such as that of a form handler.
Relationship between a domain name and the brand/organisation name of the website	Domains often relate to the brand names	Domains do not relate to the brand names
Presence of a website's domain name in non-standard positions in the URL	It is not standard practice	This is common practice in phishing URLs
Use of numerical digits or IP addresses for a domain name	It is not a common practice	It is a common practice
Encoding domain names	It is not a common practice	It is a common practice
URL length	Often short	Often long
Use of free domains managed by free web hosting services	Those owned by established organisations are not expected to use free domain names	There is a growing number of phishing websites using free domain names
Use of shortened URLs	PDC web pages are not expected to use shortened URLs	There is a growing trend of phishing PDC web page using shortened URLs
Domain name lifespan	Most of the established organisations are expected to have been using the domain name for a long time	Phishers often use their registered domain for short periods to avoid being tracked
Use of digital (TLS) certificates	Most domain names are expected to use digital certificates, especially EV certificates.	Most of the domain names still do not use the certificates. For those using them, OV and DV types are the common ones.
Number of websites sharing a host	Sharing of a host is less expected in the majority of websites	Many phishing websites sharing a host is a normal practice

4 Prediction model design

In this section, we describe how we derived our proposed features for predicting phishing PDC web pages. We also describe and illustrate the proposed system architecture of our prediction model.

4.1 Phishing web page prediction features

First, we describe how we identified PDC web pages, the web pages that collect users' personal data. As explained previously, not every web page with an HTML form or a dialogue window collects personal data. From our observations, PDC web pages usually consist of at least one word or phrase (we term as PDC phrase) in their structure and contents which is related to the specific personal data being collected. To determine common PDC phrases used by PDC web pages, we investigated 100 samples of phishing and legitimate web pages capturing the data from which we obtained a list of 43 PDC phrases (indicated in Table 3). The importance of differentiating PDC from non-PDC web pages is that we avoid predicting web pages which do not pose any phishing threat. This will avoid degrading of user's experience when accessing the non-PDC web pages and the potential false positives on these web pages which will prevent users from accessing them, causing significant implications to users and websites' owners (e.g., denial of services and losses of revenues). The list, however, is not exhaustive as there could be other PDC web pages which capture personal data not related to the PDC phrases in the list. In this case, such web pages will be regarded by our model as non-PDC web pages and thus will not be considered for the prediction analysis. To expand the list, a larger set of PDC web pages can be used to extract the phrases. For instance, one can collect a comprehensive set of known phishing PDC web pages from various phishing blacklists and algorithmically extract label and default values of all input fields, and the name attribute of the submit button in each web page to create the list.

Table 3 Common PDC phrases used in PDC web pages

Username	Login	Forgotten your password	Customer number	Log in with Facebook
User	Password	Reset password	Membership number	Log in with Twitter
Email	PIN	Debit card number	Billing information	Log in with Google
Account	Secret key	Credit card number	Billing address	Sign in with Facebook
Account number	Security code	Card number	Cardholder	Sign in with Twitter
ID	Security key	Account number	Expiry date	Sign in with Google
Sign in	Security number	Security number	Date of birth	Create an account
Sign up	Forgot password	Passcode	Birth date	
Log in	Forgot	Remember me	Phone	

Table 4 Summary of the proposed features

Feature #	*Feature name*	*Feature description*	*Feature category*	*Novel or adopted?*
1	Domain identity on a web page	A number of times a URL's domain appears in the web page structure and contents.	Web page structure and contents	Novel
2	Domain identity in copyright	Domain in a URL is checked if it matches the copyright information in the contents or not. If the two are mismatched, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Novel
3	Domain in canonical URL	Domain in a URL is compared against a common domain retrieved from canonical URLs. If the two are mismatched, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Novel
4	Domain in alternate URL	Domain in a URL is compared against a common domain retrieved from alternate URLs. If the two are mismatched, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Novel
5	Foreign domains in links	Domain in a URL is compared against a common domain retrieved from all non-object hyperlinks. If the two are mismatched, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Novel
6	The proportion of void and same web page links	The ratio of the sum of the number of void (empty) and the number of links pointing to the same web page divide by the total number of all non-object links.		Novel
7	Foreign form handler	The domain of a form handler of a web page is compared against a domain in URL and a common domain in non-object links.		Novel
8	Encoded hostname	The presence of % followed by two hexadecimal digits in the hostname is checked. If found, the web page is flagged as a phishing one otherwise it is a legitimate web page.	URL structure	Adopted
9	Encoded URL path	The presence of % followed by two hexadecimal digits in the URL path is checked. If found, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Adopted
10	Use of @ character in a URL	The presence of @ character or its equivalent hexadecimal number (%40) in the URL path is checked. If found, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Adopted
11	Domain out of position	The characters *http://*, *https://* and *www* characters and generic or country code top level domain are checked if they have been used more than once in a URL. If not, their positions in the URL will be determined if they are different from the standard ones. If any of the condition is true, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Novel

Table 4 Summary of the proposed features (continued)

Feature #	*Feature name*	*Feature description*	*Feature category*	*Novel or adopted?*
12	# of dots in hostname	The number of dots in a hostname is counted.		Novel
13	# of dots in the URL path	The number of dots in a URL path is counted.		Novel
14	Non-standard port number	For a URL that has used a port number, the number is compared against its http protocol. If the number is not 80 for http and 443 for https, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Adopted
15	# of obfuscation characters in the hostname	Number of ‘_’, ‘-’ and ‘=‘ characters in a hostname is counted.		Novel
16	# of obfuscation characters in the URL path	The number of ‘_’, ‘-’ and ‘=‘ characters in a URL path is counted.		Novel
17	# of forward slashes	The number of ‘/’ in a URL is counted.		Adopted
18	# of characters in the hostname	The total number of characters in a hostname is counted.		Adopted
19	# of characters in the URL path	The total number of characters in a URL path is counted.		Novel
20	The IP address in a hostname	The presence of an IP address in a hostname is checked. If found, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Adopted
21	Numeric in a hostname	The number of numeric characters in a hostname is counted.		Novel
22	Numeric in a URL path	The number of numeric characters in a URL path is counted.		Novel
23	Shortened URLs	The use of shortened URL is checked by comparing a hostname of the URL against a list of 242 hostnames of the collected shortening URL services. If found, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Adopted
24	Free domain services	The use of a free domain is checked by comparing a domain of a web page domain against a list of domains of the most abused free domain services we compiled from Anti-Phishing Working Group (APWG)'s reports on global phishing surveys between 2008 and 2017.		Novel
25	Domain validity	An expiry date of a web page's domain registration (from WHOIS database) is compared with the current date to check if it is still valid or not. If it is overdue, the web page is flagged as a phishing one otherwise it is a legitimate web page.	WHOIS records	Novel

Table 4 Summary of the proposed features (continued)

Feature #	*Feature name*	*Feature description*	*Feature category*	*Novel or adopted?*
26	Form handler's domain validity	An expiry date of a form handler's domain registration (from WHOIS database) is compared with the current date to check if it is still valid or not. If it is overdue, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Adopted
27	Domain age	A difference between the current date and the web page domain's first date of registration (from WHOIS database) is computed.		Adopted
28	Form handler domain's age	A difference between the current date and the form handler domain's first date of registration (from WHOIS database) is computed.		Adopted
29	Type of SSL certificate	The type of SSL certificate used by the web page's domain is determined.	TLS certificate	Novel
30	Domain, certificate and geolocation country matching	Country names in the ccTLD (for URLs with ccTLDs only), SSL certificate and location of the host are compared. If they do not match, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Novel
31	URL ranking in search engines	A URL is searched in the Google and Bing search engines. URLs in the top five results returned by each engine are compared against the searched URL. If none of the results are matching, the web page is flagged as a phishing one otherwise it is a legitimate web page.	Web page reputation	Novel
32	Hostname ranking in search engines	A hostname is searched in the Google and Bing search engines. Hostnames in the top five results returned by each engine are compared against the searched hostname. If none of the results are matching, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Novel
33	Domain ranking in search engines	A domain is searched in the Google and Bing search engines. Domains in the top five results returned by each engine are compared against the searched domain. If none of the results are matching, the web page is flagged as a phishing one otherwise it is a legitimate web page.		Novel
34	Counts of matched hostname's IP address in a blacklist	The number of times an IP address of a hostname appears in a list of IP addresses of blacklisted phishing URLs is counted. A 3-month-old data of a blacklist is used.		Novel
35	Counts of matched domain's IP address in a blacklist	The number of times an IP address of a domain appears in a list of IP addresses of blacklisted phishing URLs is counted. A 3-month-old data of a blacklist is used.		Novel

Based on the differences in the structural characteristics between phishing and legitimate websites described in section 3, we derived various potential features for distinguishing phishing PDC web pages from legitimate ones. We also studied features used by previous works addressing the same problem and identified those which can be extended or improved, based on the mentioned characteristics, to add to our set of features. In addition, we adopted some of the features, in our proposed set of potential features, those which were used and defined as strong predictors in several works. To identify features which are strongly exhibited by the current phishing PDC web pages and thus are likely to be useful predictors, we investigated them in the same 100 samples of phishing and legitimate PDC web pages mentioned above. This was done by algorithmically analysing the occurrence patterns of values of each feature across the two sets of web pages. The features whose patterns of values were more consistent in one set compared to the other were considered to be potential predictors. For example, we counted the number of URLs from each set that contained the character @ in their strings (feature #10 in Table 3). We found that the character appeared in almost 18% of all the phishing URLs while none were in the legitimate URLs. From this investigation, 35 of such features were identified. We categorise the features into five groups namely web page structure and contents, URL structure, WHOIS records, TLS certificate and web page reputation. The categorisation is based on the similarity of sources of the features. For instance, all features which were derived from the character composition of a URL string are grouped as URL structure. Of the 35 features, 24 are new ones introduced by this study and 11 features are adopted from previous related works. In this section, we describe some of the features which were observed to be among the best features for this problem. Table 4 below summarises all the proposed features along with their descriptions.

4.2 System architecture of the prediction model

Training process

Our prediction model based on the proposed features earlier is built using the following six-step process (illustrated in Figure 1 as steps 1 to 6):

Step 1 Collection of known phishing and legitimate PDC web pages

A set of each type of web pages is collected from its respective database and then labelled as phishing or legitimate accordingly. In this study, we collected active phishing web pages from a phishing blacklist while legitimate web pages were collected from a ranked list of the most visited websites.

Step 2 PDC web page filtering

The model is aimed at analysing only PDC web pages. This module, therefore, determines if a web page consists of an HTML form or a JavaScript pop-up window and at least one of the PDC phrases as described in Section 4.1.

Step 3 URL redirections check

Some of the web pages are designed to perform one or more URL redirections before landing to their actual URLs. We need to obtain the final redirected URL of each web page in order to collect relevant URL based features. Checks are carried out for redirections embedded in the web page structure and those provided through URL shortening services. The former is indicated by the presence of a URL in the meta tag's refresh attribute in the head section of the web page or the JavaScript's window location attribute. In the latter, the shortened URL based redirections are determined by comparing the web page's hostname against a list of known shortening URL providers we collected. If a match is found, short to long URL conversion is performed using Untiny's online converter (http://untiny.com/).

Step 4 Feature extraction

All the features of a PDC web page are extracted from various local and external sources to build a training dataset in step 5.

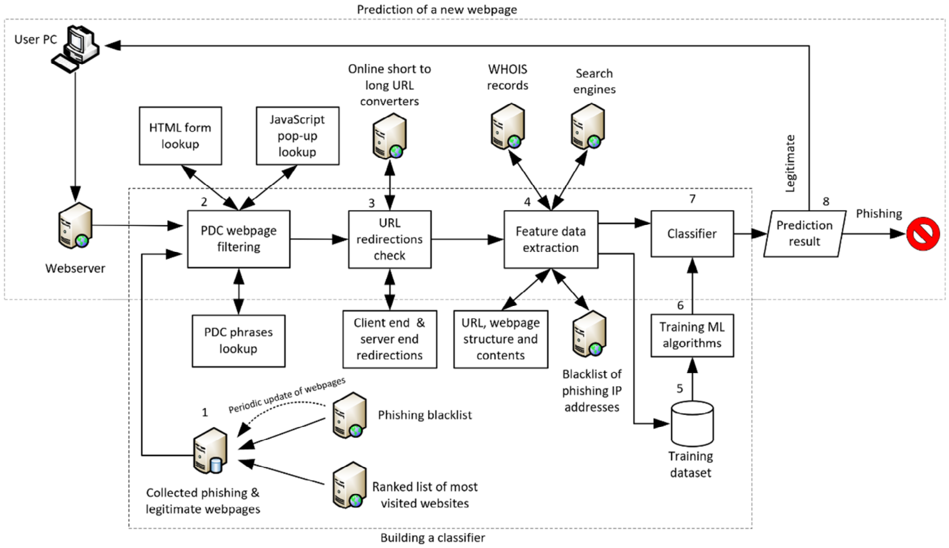
Step 5 Training a classifier

Use the training dataset to train an ML algorithm to build the classifier.

Prediction process

The process of predicting a new web page requested by a user is shown in steps 2 to 8 in Figure 1. The web page is retrieved from a web server after being requested by the user. A check (2) is performed to establish whether it is a PDC web page or not. If it is not, it is passed to the browser and displayed to the user. If it is a PDC web page, any redirections are resolved (3). Its features are extracted (4) and passed as input to the classifier which makes a prediction (7–8). If the web page is classified as phishing, the user's access to the web page is blocked or warned otherwise it is permitted. The designs of phishing web pages are likely to change over time as phishers adapt their methods to evade detection. We, therefore, propose periodic addition of new phishing web pages and re-training of the classifier to ensure the classifier always provides an optimal performance.

Figure 1 The system architecture of the proposed model for predicting zero-day phishing PDC web pages (see online version for colours)



5 Experiments and results

We designed several experiments to build and evaluate a binary classifier that predicts phishing PDC web pages. We ran two sets of experiments with the aim of identifying the most accurate set of features and the best performing ML algorithm for the classifier. In the first set, 12 traditional ML algorithms, comprising six standard ML algorithms and six ensemble algorithms, are used for evaluation. The former includes linear regression (LR), k-nearest neighbour (k-NN), decision tree (DT), naive Bayes (NB), SVM and artificial neural network (ANN) (Brownlee, 2016a, 2016b; Chauhan, 2020; Dickson, 2019; Müller and Guido, 2017; Navlani, 2019; Nicholson, n.d.; Ray, 2017; VanderPlas, 2017). The latter are random forest (RF) (Yiu, 2019), gradient boosting (GB) (Brownlee, 2021a), LightGBM (LGBM) (Ke et al., 2017), XGBoost (XGB) (Brownlee, 2016a), extra trees (Brownlee, 2021b) and CatBoost (Hancock and Khoshgoftaar, 2020). In the second set of experiments, we evaluated the features using 3 DL algorithms namely fully connected feedforward deep neural networks (FC-DNN), long short-term memory (LSTM) and one dimensional convolutional neural network (1D CNN) (Al-Garadi et al., 2018; Apruzzese et al., 2018; Berman et al., 2019; Dertat, 2017; Kiranyaz et al., 2021; Moolayil, 2019; Nguyen, 2018; Phi, 2018; Verma, 2019). We use eight standard ML performance metrics (described in Section 5.3 below) to compare the performance results of the algorithms.

The ML experiments were runs on a machine with MS Windows Home, 16 GB memory and Intel's i7 processor specifications. DL based experiments were run on Google's Collaboratory platform. We developed and used a Python v3.6 application to extract and pre-process data, create a training dataset, and train and evaluate the algorithms. The extracted data was stored in the MySQL database.

5.1 Training datasets

We collected 13,494 legitimate and 12,621 phishing PDC web pages to build a training dataset. To obtain legitimate web pages, we first collected more than 100,000 top websites from a ranked list of 1 million most visited websites from Tranco (https://tranco-list.eu). Using the hostname of each website combined with each of the PDC phrases listed in Table 3 at a time, we searched for candidate PDC web pages related to these websites in the Google and Bing search engines. We extracted the maximum possible number of URLs returned by each search, downloaded the web page of each URL and then checked (using the PDC web page filtering procedure described in Section 4.2) whether it prompts for personal data or not. Finally, features of each of the confirmed PDC web page were extracted using our application and added to the MySQL database.

We obtained a list of phishing PDC web pages from an online repository of the confirmed phishing URLs managed by PhishTank (https://www.phishtank.com). The database is one of the most reliable sources of blacklisted phishing URLs in the cybersecurity domain. Since the database is updated hourly, we downloaded its list four times a day over five days. In each list, we retrieved each active URL, downloaded its web page and then checked whether it is a PDC web page or not. We then extracted the features of all the qualified web pages and recorded them in the database. In several cases, multiple URLs were observed to belong to the same hostname, hosting the same or different PDC web pages. To avoid the possibility of the excessive influence of a few hostnames leading to a biased model, we limited each hostname to at most 20 unique URLs.

Table 5 A summary of the collected data used to build the training dataset

PDC web page type	*Source*	*Size*
Legitimate	Tranco's list of most visited websites, Google and Bing search engines	13,494
Phishing	PhishTank online repository	12,621
		26,115

5.2 *Data pre-processing*

We applied several standard data pre-processing techniques to transform raw data into a form that ML algorithms can efficiently learn to produce optimal prediction results. We first identified features with missing values, as summarised in Table 6. There is no standard threshold percentage to determine whether a feature with missing data should be used or not. For this study, we set a threshold of 50%, which is commonly used by many practitioners (Kunal, 2015; Madley-Dowd et al., 2019). We, therefore, dropped features # 4 and 3 as they exceeded the threshold. We also analysed correlations between the features using Pearson's correlation matrix (Brownlee, 2020; Sillipo and Widmann, 2019) in order to determine redundant features. We dropped features # 24, 25 and 30 because they have correlation values of 1.0 with features # 22, 23 and 29 respectively. We then encoded all categorical features as unique numeric values, with missing values given their unique labels (Pathak, 2018).

Four imputation methods (mean, median, most frequent and k-NN (k = 4), which are commonly used in replacing missing values in numerical features, were compared. We found that mean imputation produced the best performance when we ran one of the algorithms (RF) on the dataset and therefore it was used to replace the missing values. Finally, we applied a data scaling technique (Brownlee, 2016b) to standardise the data ranges of all the features by transforming the data in each feature such that its distribution has a mean value of 0 and a standard deviation of 1.

Table 6 Features with missing values

Feature #	*Feature name*	*% missing values*
4	Domain in alternate URL	91.4
3	Domain in canonical URL	87.2
2	Domain identity in copyright	48.0
26	Type of TLS certificate	45.4
27	Domain, certificate and geolocation country matching	42.3
22	Domain validity	14.2
23	Domain age	14.1
24	Form handler's domain validity	14.1
25	Form handler domain's age	14.1
6	The ratio of void and same web page links	1.9

5.3 Performance results

We use various evaluation measures to report the prediction performance of the classifiers. These are accuracy, false positive rate (FPR), false negative rate (FNR), precision, recall, F1-score, ROC curve and area under curve (AUC) (Brownlee, 2014, 2018; Müller and Guido, 2017). They are defined as follows:

$$Accuracy = TP + TN / TP + TN + FP + FN$$

$$FPR = FP / FP + TN$$

$$FNR = FN / FN + TP$$

$$Precision = TP / TP + FP$$

$$Recall = TP / TP + FN$$

$$F1\text{-}score = 2 * Precision * Recall / Precision + Recall$$

The above performance measures are derived from the counts of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Note that if an instance is positive and it is classified as positive, it is defined as TP. If the instance is negative and it is classified as positive, it is FP. A negative instance classified as negative is TN and if it is classified as positive, it is called FN. A positive instance in this problem is the phishing PDC web page. We present and compare the results of individual classifiers for both traditional ML and DL experiments in the following subsections.

5.3.1 Results of traditional machine learning algorithms

The 12 traditional ML algorithms were run and their results were compared to identify the best performing algorithm for the classifier. First, automated feature selection using a recursive feature elimination method (Sillipo and Widmann, 2019) with cross validation and RF algorithm was performed which identified a subset of 26 features to be the best features for the classifier. A stratified cross validation technique (*k*-fold where *k* is 10) (Brownlee, 2016b; Müller and Guido, 2017) was applied to train and test the algorithms in order to obtain their average prediction scores. As summarised in Table 5, the dataset composition by classes is nearly balanced, thus class balancing techniques were not applied. Table 7 summarises the results of the untuned algorithms and Figure 2 shows the performances of ML algorithms across all threshold values in their ROC curves. The results indicate that CatBoost has the highest accuracy, the lowest FPR and the second lowest FNR, thus, identified as the best algorithm for the implementation of the classifier. The CatBoost was then tuned using a random search method (Worcester, 2019) to optimise its performance. The tuning of CatBoost yielded an *accuracy of 99.02%*, *FPR of 0.90%* and *FNR of 1.03%*. Values of hyperparameters of the tuned CatBoost are indicated in Table 8.

Table 7 Performance results of the traditional ML algorithms

Algorithm	*Accuracy (%)*	*FPR (%)*	*FNR (%)*	*Precision*	*Recall*	*F1 score*	*AUC*
LR	94.87	3.71	1.48	0.96	0.93	0.95	0.98
K-NN	95.72	2.91	5.74	0.97	0.94	0.96	0.98
DT	97.42	2.59	2.57	0.97	0.97	0.97	0.97
NB	78.48	1.33	43.11	0.98	0.57	0.72	0.96
SVM	96.00	2.48	5.63	0.97	0.94	0.96	0.99
ANN	96.94	2.18	3.99	0.98	0.96	0.97	0.99
RF	98.45	1.13	2.00	0.99	0.98	0.98	1.0
GB	97.89	1.48	2.77	0.98	0.97	0.98	1.0
LGBM	98.55	1.05	1.90	0.99	0.98	0.98	1.0
XGB	98.71	1.09	1.65	0.99	0.98	0.99	1.0
ExtraTrees	98.49	1.12	1.68	0.99	0.98	0.98	1.0
CatBoost	*98.89*	*1.03*	*1.56*	*0.99*	*0.98*	*0.99*	*1.0*

Figure 2 ROC curves of the trained traditional ML algorithms (see online version for colours)

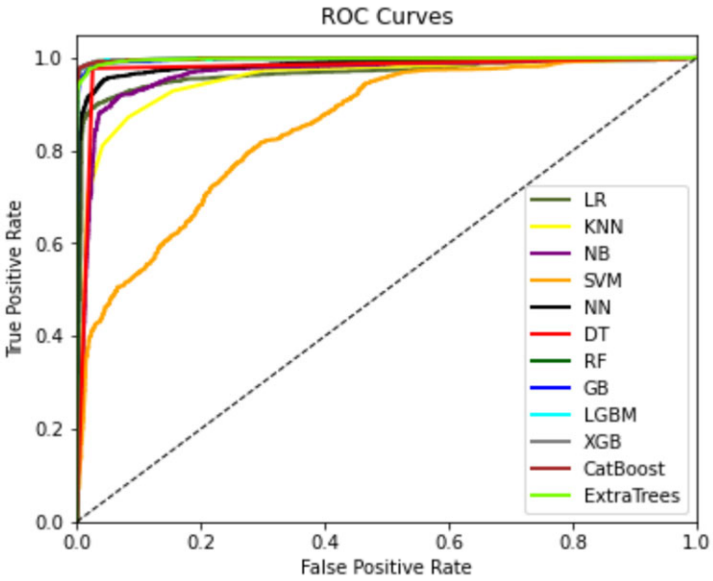


Table 8 Optimal values of hyperparameters of the tuned CatBoost

Parameter	*Description*	*Value*
iterations	Number of times the model evaluates the validation data using the loss function before updating its parameters	1,100
learning_rate	The rate at which the model updates its parameters to improve its validation error	0.1
depth	Number of levels in each decision tree	8
subsample	The subsample ratio of columns when constructing each tree	0.4
l2_leaf_reg	L2 regularisation term on weights	9

5.3.2 Results of DL algorithms

The three DL algorithms were run on the same dataset as the previous algorithms to evaluate the features. First, the training dataset for LSTM and 1D CNN was converted into a sequential shape, a standard input data format for the two algorithms, with a time step assigned to 1. The algorithms were then tuned by a random search method. We first identified key hyperparameters for tuning and their considerable range of values for evaluation. In each hyperparameter, we identified a set of considerable values for performance tuning (indicated in Table 9). We also attempted to tune with multiple hidden layers. We found that only one hidden layer was sufficient to produce optimal performance in each algorithm. Additional layers did not improve the performances. The identified optimal values of all the hyperparameters were then used to build the classifiers. The final result of each classifier was obtained by taking an average of the performances of five runs of the tuned classifier. Table 10 summarises the performance results of the tuned algorithms. Figures 3(a)–3(c) illustrate network architectures of the tuned DL algorithms along with the tuned hyperparameters and their optimal values.

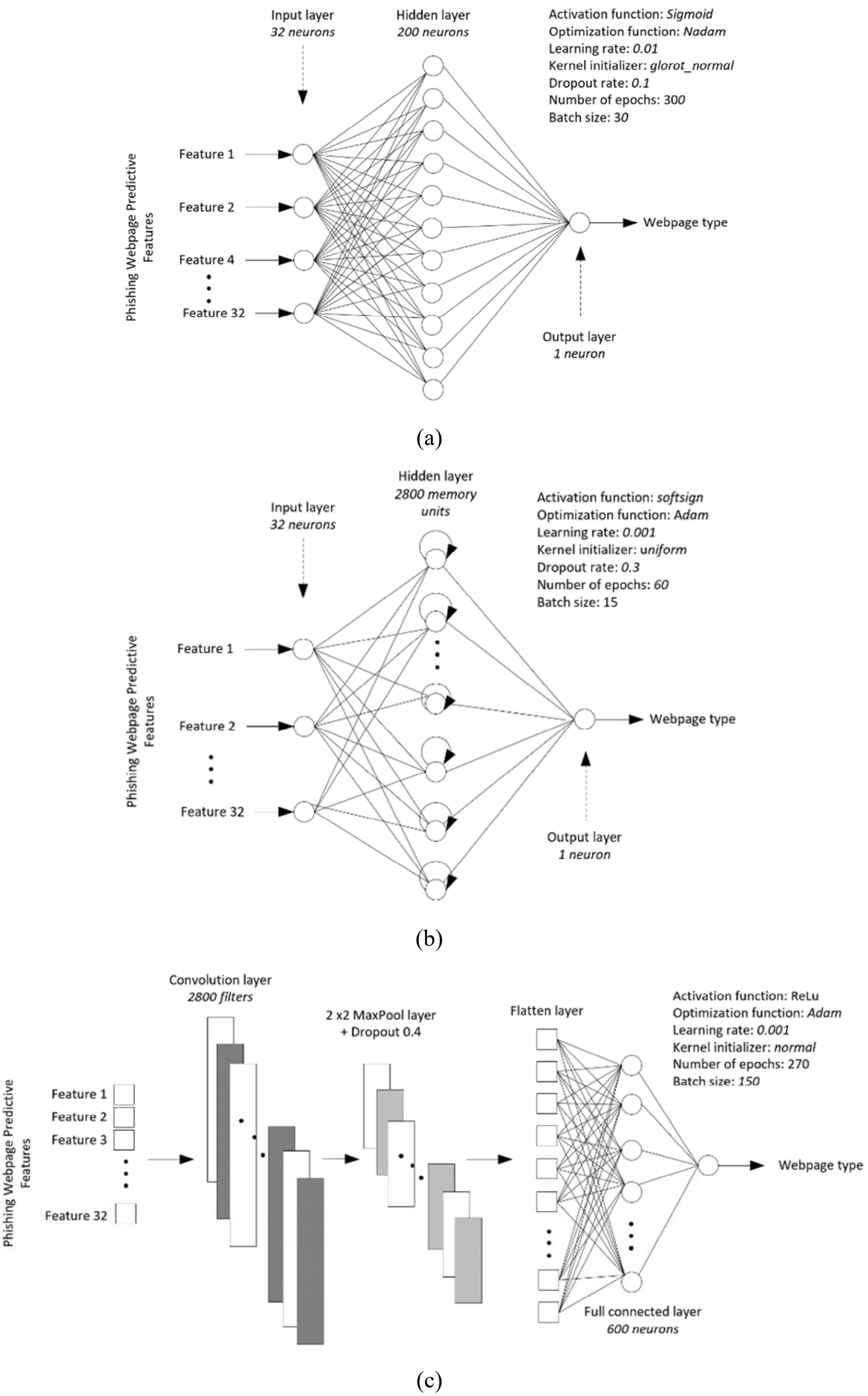
Table 9 Hyperparameters and their ranges of values evaluated for tuning the three DL algorithms

Hyperparameter	*Range of evaluated values*
Number of neurons in dense layers of FC-DNN/memory units in a hidden layer of LSTM/filters in a convolution layer of CNN	10, 30, 50, 80, 100, 150, 200, 300, 400, 600, 800, 1,000, 1,200, 1,400, 1,600, 1,800, 2,000, 2,200, 2,400, 2,800, 3,000
Activation functions	Relu, tanh, sigmoid, hard_sigmoid, linear, softmax, softplus, softsign
Optimisation algorithms	SGD, RMSprop, Adagrad, Adadelta, Adam, Adamax, Nadam
Learning rates	0.001, 0.01, 0.1, 0.2, 0.3
Kernel initialisers	Uniform, lecun_uniform, normal, zero, glorot_normal, glorot_uniform, he_normal, he_uniform
Dropout rates	0.1, 0.2, 0.3, 0.4, 0.5
Batches	15, 30, 50, 70, 90, 110, 130, 150
Epochs	10, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300

Table 10 Performance results of the DL algorithms

Algorithm	*Accuracy (%)*	*FPR (%)*	*FNR (%)*	*Precision*	*Recall*	*F1 score*	*AUC*
FC-DNN	*97.28*	*2.13*	*3.33*	*0.98*	*0.97*	*0.97*	*0.97*
LSTM	95.71	3.50	5.11	0.96	0.95	0.96	0.99
CNN	95.66	3.14	5.61	0.97	0.94	0.95	0.98

Figure 3 (a) FC-DNN architecture of the classifier with optimal values of the tuned hyperparameters (b) LSTM architecture of the classifier with optimal values of the tuned hyperparameters (c) 1D CNN architecture of the classifier with optimal values of the tuned hyperparameters



5.3.3 Overall results and feature analysis

We found that the tuned CatBoost achieved the highest performance of all algorithms across most metrics. Except for the NB, all the algorithms, however, exhibited good performance values in all metrics. Given that the algorithms use different assumptions to develop their prediction rules, the observed good results show that our features are generally effective in predicting phishing web pages. It is interesting to see that a number of traditional ML algorithms, mostly the ensembles, have outperformed the DL algorithms. Recent studies have shown that the later tend to outperform the former on prediction tasks involving unstructured data such as texts and images. However, Shwartz-Ziv and Armon (2022) empirically illustrated that ensemble algorithms tend to outperform DL algorithms in most classification problems involving structured/tabular data. Given that our data is also structured/tabular, it is not surprising that we also observe the same trend. In addition, traditional algorithms tend to perform well in small and medium size datasets while DL algorithms perform well in large datasets (Moolayil, 2019). Our dataset is relatively smaller compared to some of those experimented by Shwartz-Ziv and Armon (2022), thus, it was unlikely to expect a different pattern.

Table 11 breaks down the features in the full set (35 features) and the best set (26 features) by category. There is a high representation of features from each category in the best set, with the exception of WHOIS records, which has only 1 out of 4 in the best set. This indicates that all the categories are important in the prediction though to different extents. We also compared the performances of the two feature sets using the CatBoost algorithm to justify the use the best feature set. Figures 4(a)–4(b) show the differences between the two sets in terms of accuracy and error rates. Clearly, feature selection approach is important in this problem as it makes a significant difference. Using the same algorithm, we evaluated the performances of the best feature set belonging to a specific category only. The results are shown in Figures 5(a)–5(b). Web page reputation, URL structure, and web page structure and contents categories produced the highest accuracy and lowest FPR rates whereas WHOIS and TLS certificate categories achieved the lowest accuracy rates and the highest FPR rates. The former, therefore, are the strongest predictors while the latter are the weakest. Though we replaced all the missing values with the imputed ones, the high percentages of missing values of the WHOIS and TLS certificate related features (as indicated in Table 6) is likely to be the main reason why the two categories produced poor performances.

Table 11 Distribution of feature categories in the set of the 26 best features

Feature category #	*Feature category*	*Tally of features in the full set*	*Tally of features in the best set*	*Best features # (see Table 4 for feature #)*
1	Web page structure and contents	7	5	1–2, 5–7
2	URL structure	17	14	9–19, 22–24
3	WHOIS records	4	1	27
4	TLS certificate	2	2	29–30
5	Web page reputation	5	4	31–32, 34–35

Similarly, we evaluated the performance contributions of the novel and existing features in the set of best features and compared them against the overall set of best features [see

Figures 6(a)–6(b)]. While our novel features achieved better results compared to the existing features, a combination of the two increased the overall accuracy and lowered the error rates, especially the FNR. This suggests that their combination is important for achieving optimal prediction performances, as in the case of feature categories. Increasing the diversity of the features is also likely to have a benefit in terms of hardening the solution against detection evasion techniques.

Figure 4 (a) Accuracy rates of the feature categories (b) Error rates of the feature categories (see online version for colours)

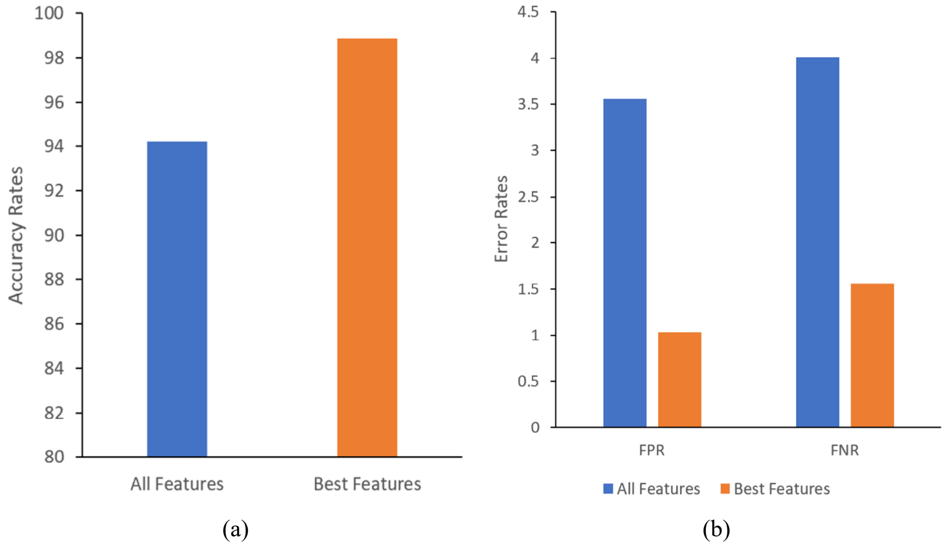


Figure 5 (a) Accuracy rates of the feature categories (b) Error rates of the feature categories (see online version for colours)

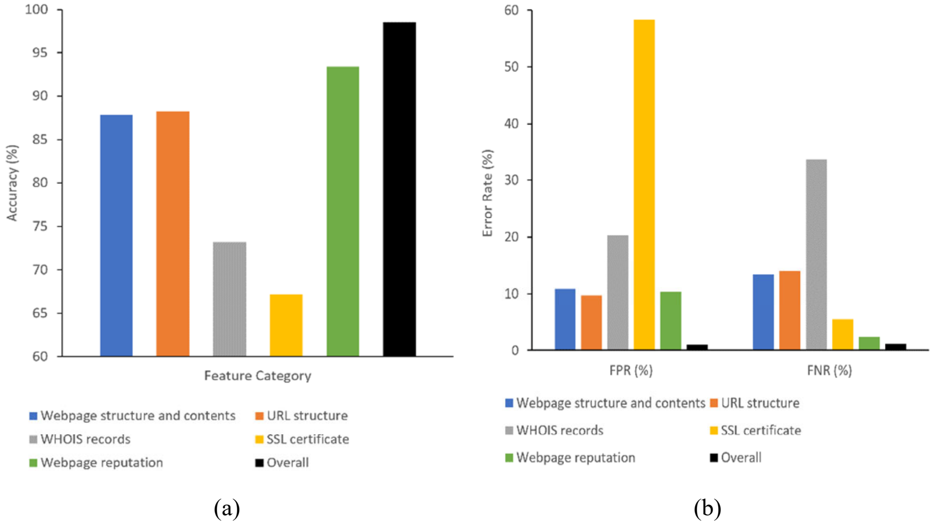
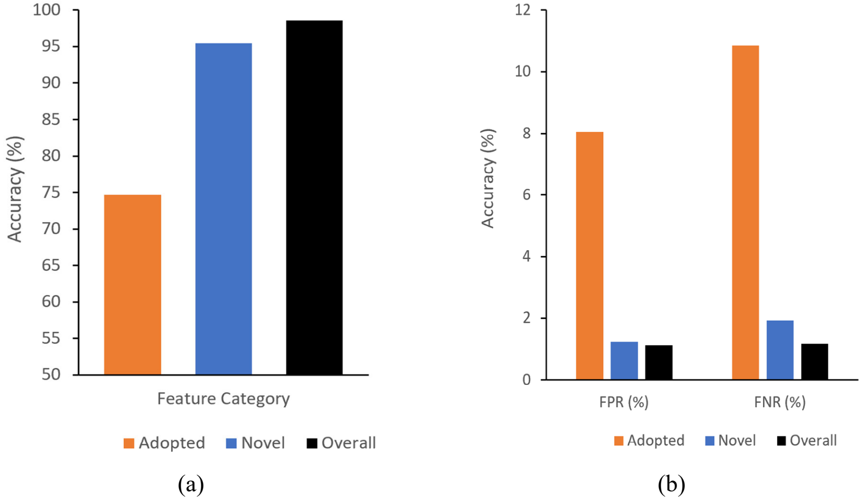


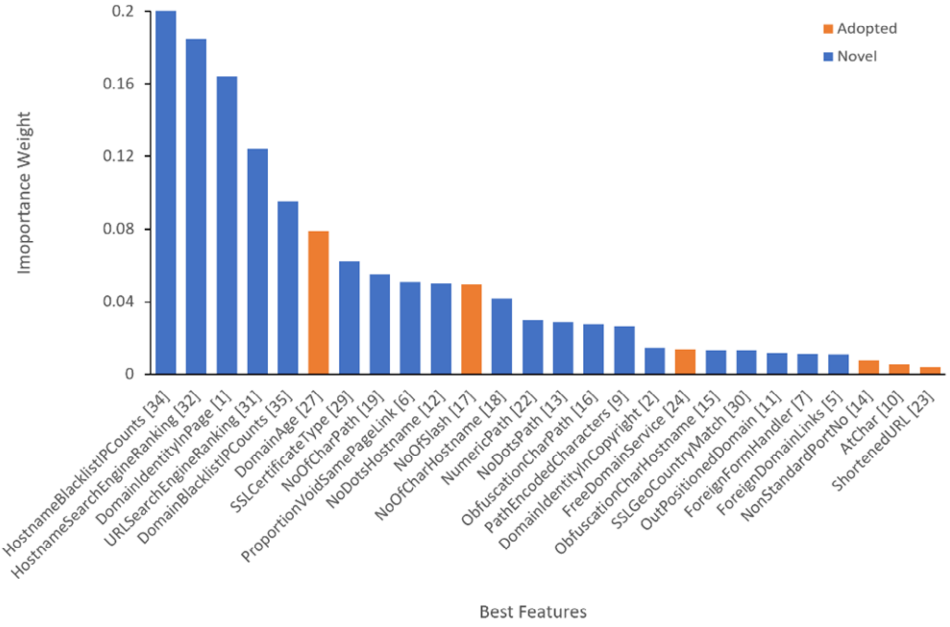
Figure 6 (a) Accuracy rates of the existing and novel features (b) Error rates of the existing and novel features (see online version for colours)



(a)

(b)

Figure 7 Ranking of the best features by importance weight (see online version for colours)



Note: The numbers in the brackets represent the numbers of features as indicated in Table 3.

To evaluate the individual features, feature importance weights of the best features were computed using the tuned CatBoost to determine the prediction strength of each feature relative to others. Figure 7 shows the ranking of the 20 novels and six existing features in terms of their importance weights. The feature with the largest weight indicates the strongest predictor while the one with the lowest weight is the weakest predictor. As a general observation, the novel features rank higher than the adopted ones. The top 5 features are the novel ones while the last 3 are the existing ones. Similarly, 6 of the 9 third-party-based features are ranked higher compared to most of the 17 local features (those based on URL structure, and web page structure and contents). 6 of the top 7 features are third-party-based features. This indicates that the novel and third-party-based features are more effective for the prediction. Most of the web page reputation-based features are the strongest predictors while the weakest ones consist mostly of the URL structure-based features.

The highest ranked feature is feature 34 (see Table 4), which is related to the number of times the hostname's IP address matches with the IP addresses of blacklisted phishing websites. The feature's data distribution shown in Figure 8 (presented using boxen plots) explains this by showing that hosts of unknown phishing hostnames match with a large number of hosts previously known to host backlisted phishing websites while only a small number of legitimate hostnames do the same. The distribution suggests that many phishers use a small number of machines to host multiple phishing websites at different times. Meanwhile, the small number of hosts of legitimate websites that were matched indicates that phishers also use compromised legitimate hosts to do the same. This feature and feature 35, which is related to the number of times the domain's IP address matches with IP addresses of blacklisted phishing websites, have a moderate correlation value of 0.6, showing that there is a medium level of correlation between them. This, combined with the difference in ranking between the two features, suggests that phishers, in some cases, host their hostnames and domains on different machines. We confirmed this trend in our dataset by observing that some of the phishing web pages have different counts for features 34 and 35. We think phishers use the approach to limit the impact of detection through blacklisting, that is, if a host of the hostname is taken down, the domain can still operate.

Hostname matched in a search engine's top 5 results (feature 32) is the second ranked feature. As shown in Figure 9, about 90% of the phishing hostnames are not returned in the search results, suggesting either that they were not indexed because their web pages were recently created, or that the web pages did not meet the search engine's high ranking indexing criteria. Conversely, about 90% of the legitimate hostnames were returned in the search results. Counts of a domain identity appearing in a web page structure and contents (feature 1) appear in 3rd place and is the highest ranked feature based on web page structure and contents. Figure 10 shows that a larger number of legitimate URLs contain domain identities which are appearing multiple times on a web page compared to those of phishing URLs. This confirms that many phishing web pages exhibit a mismatch between the domain names in their URLs and the identities of the organisations the web pages appear to belong to, thus, most of them are being hosted in the domains registered by attackers.

Domain age (feature 27) takes the 6th position and it is the only feature from the WHOIS records category in the ranking. Its distribution of data (see Figure 11) shows that phishing domains have shorter domain ages, with the majority of them having ages between 1,000 and 4,000 days and with a median value of just below 2,000 days, while legitimate domains have longer domain ages, in which the majority have ages between 3,500 to 8,000 days and a median value of just below 6,000 days. The observed domain ages of the phishing domains are still significantly longer than the expected ones reported in Akamai (2019). This suggests that attackers have generally increased the duration of their domains staying active, possibly for evading detection techniques based on domain age.

Figure 8 Data distribution of counts of hostname's IP address matching with phishing blacklisted IP addresses (see online version for colours)

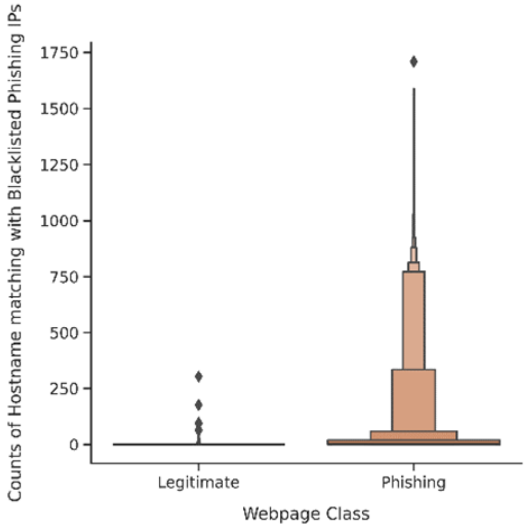


Figure 9 Data distribution of the matching of hostnames in a search engine's results (see online version for colours)

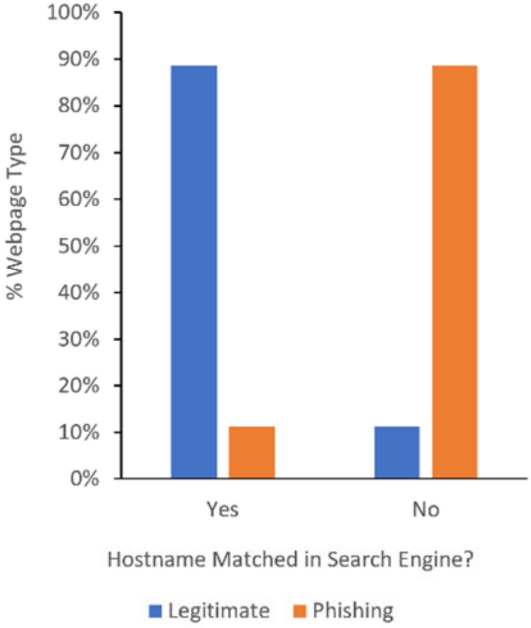


Figure 10 Data distribution of counts of domain identity appearing in the web page structure and contents (see online version for colours)

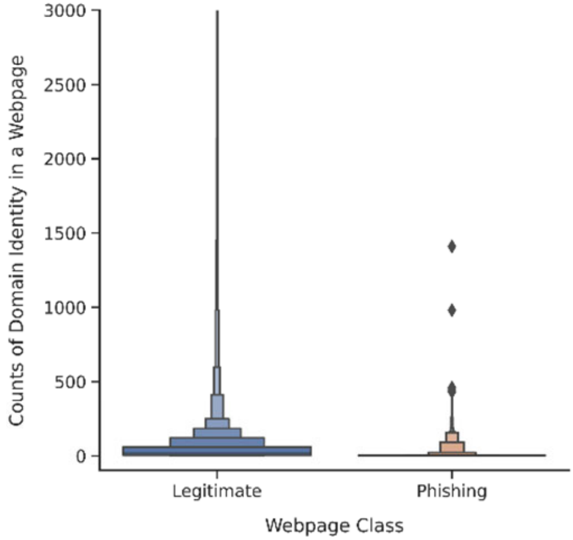
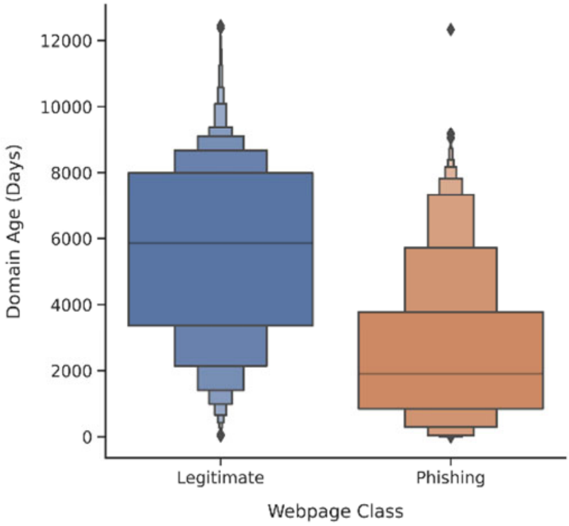


Figure 11 Data distribution of domain age (see online version for colours)



The TLS certificate type (feature 29) is the 7th feature and the highest ranked of the two features in the TLS certificate category. Figure 12 shows the breakdown of certificate usage by type for the two web page classes. As expected, the majority of phishing web pages still do not use any certificate. There are more of them using DV than OV and EV, suggesting that phishers are taking advantage of the least strict validation procedures in obtaining DV certificates to try to make their web pages as legitimate as possible. We had expected EV certificates to be popular among legitimate websites due to their high security and user trust but it is the least popular category. More surprisingly, a significant percentage of legitimate websites do not use any certificate at all. This shows that most legitimate website owners are yet to understand the significance of using EV in their websites in improving security and users' trust.

Figure 12 Data distribution of types of TLS certificate (see online version for colours)

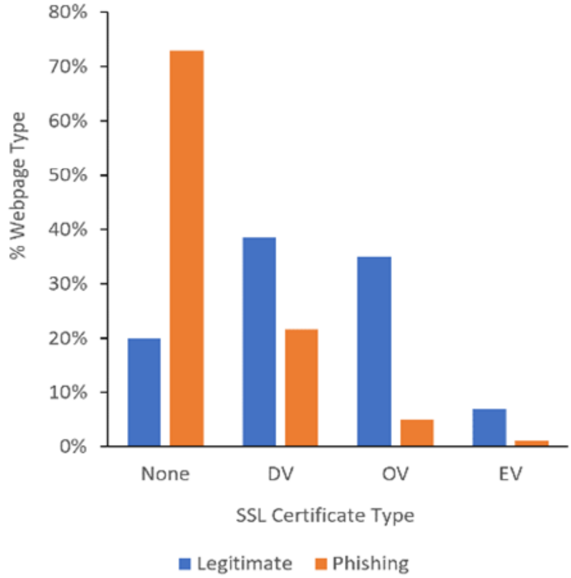
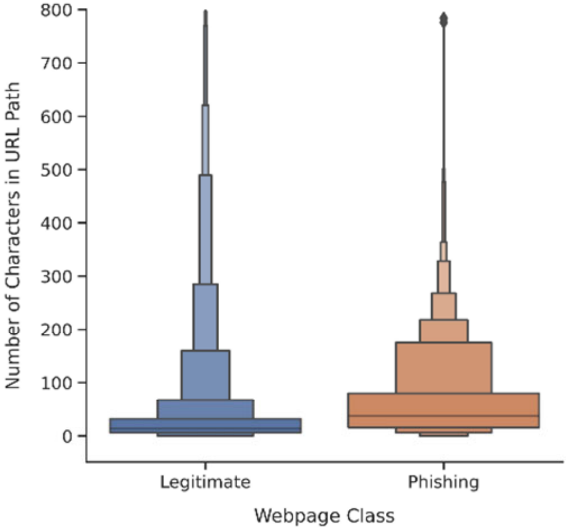


Figure 13 Data distribution of counts of characters in a URL path (see online version for colours)



The number of characters in the URL path (feature 19) takes the 8th position and is the highest ranked feature based on URL structure. From Figure 13, phishing web pages tend to have longer URL paths. This is consistent with other features related to URL paths including the number of forward slashes (feature 17), the number of numeric in a URL path (feature 22), the number of dots in a URL path (feature 13), the number of obfuscation characters in a URL path (feature 16), number of encoded characters in a URL path (feature 9) and the use of obfuscation characters in the hostname (feature 15) being among the best predictors (at 11th, 13th, 14th, 15th, 16th and 21st ranking positions respectively). This suggests that phishers obfuscate their true URLs through the addition of various characters in URL paths, which increases the length of paths. The three URL based features (features 14, 10 and 23), which were also adopted from other works, were the least ranked ones and thus were the weakest ones among the best features.

5.4 Detection time

We measured the runtimes of the model's stages (described in Section 4.2) namely retrieval of a web page from its server, PDC web page filtering, URL redirections check, feature extraction, training the dataset and prediction analysis. Table 12 summarises the average times. We only measured the feature extraction time for the 26 features in the best set. The sum of the average times of the stages in the classification process is a little under 7.2 seconds. Feature extraction is responsible for about 75% of the overall detection time. We observed that the extraction of 9 third-party and 17 local-based features take 3.05 and 2.31 seconds respectively, so the average time for a third-party feature is 0.34 seconds and that for a local feature is 0.14 seconds. Overheads in data retrieval from the third parties' servers and network overheads are likely to be the main reasons for the difference. Comparing the extraction times for each of the third-party features with the average time for a local feature, Figure 14 shows that the blacklist and the search engine-based features have the longest retrieval times. They all take longer than the average local feature. It is important to note that the runtime of each activity could be improved with more efficient Python libraries and code optimisation. Also, the features were queried and generated sequentially and the overall speed could likely be improved by introducing some concurrency.

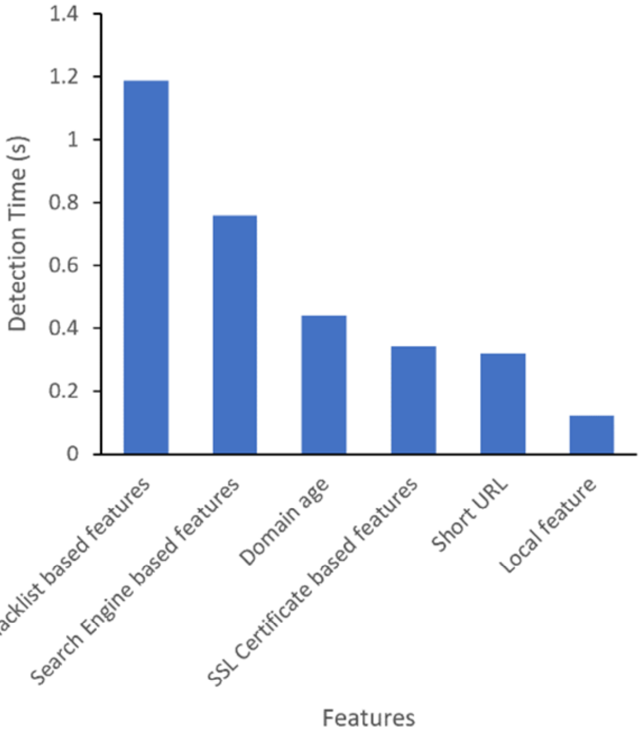
Table 12 Runtimes of the model's modules

Module	*Runtime (s)*
PDC web page loading	0.8430
PDC web page filtering	0.0976
URL redirections check	1.2959
Feature extraction	5.3600
Prediction analysis	0.0002
Total prediction time per web page	*7.1537*
Training a classifier	18.2300

5.5 Model validation using new data

Phishers are likely to adopt new ways of designing their phishing web pages over time to evade some of the detection features. To check whether the performance of the model degrades with time, we collected a new testing dataset consisting of 2,736 legitimate and 2,498 phishing PDC web pages 14 months after we collected the training dataset. After performing similar data pre-processes as described in section 5.3, the processed data was tested against the same tuned CatBoost described in section 5.3.1 using the same training dataset described in section 5.1. The classifier achieved an *accuracy of 99.05%, FPR of 1.01%* and *FNR of 1.10%*. Compared with the results reported in section 5.3.1, the model performed slightly better in terms of accuracy but slightly low in terms of error rates. The results confirm that the excellent performance achieved was not restricted to the specific dataset and that it remained effective in detecting phishing web pages after over a year. This indicates that frequent retraining will not be required to adapt to new tactics employed by phishers.

Figure 14 Comparison of extraction times of the best features (see online version for colours)



6 Discussion

6.1 Comparison with existing works

We compare the performance of our work against works which have also used ML to predict phishing web pages as a binary classification task. The comparison is done in terms of prediction performance, diversity of features, and the number of ML algorithms and metrics used for evaluation. Table 13 provides a summarised comparison chart.

Table 13 Performance comparison of some of the related works with our work

Work	*Feature # and categories*	*# feature categories*	*Data size (URLs)*	*Evaluation algorithms*	*Performance*
Ma et al. (2009)	5 WHOIS records, 9 URL, 1 network, 1 geolocation, 14 web page reputation features	5	35,500	NB, SVM, **LR**	Acc = 95%–99% Error rates = 0.9%–33.5%
Xiang et al. (2011)	7 URL, 4 web page contents, 1 WHOIS record and 3 web page reputation	4	13,064	SVM, LR, **Bayesian Network**, DT, RF, Adaboost	Acc = 92.3% FPR = 1.38% FNR = 0.95%
Shirazi et al. (2017)	30 URL, web page structure, WHOIS records and web page reputation features	4	12,000	SVM, **FC-DNN**	TP = 89% TN = 90.3% AUC = 0.9
Jain and Gupta (2018)	8 URL and 11 web page structure-based features	2	4,059	SVM, **RF**, LR, ANN, NB	Acc = 99.09% FPR = 1.25% TPR = 99.39%
Sahingoz et al. (2019)	1 web page content and 40 URL structure features	2	73,575	NB, **RF**, kNN, Adaboost, K-star, SMO and DT	Acc = 97.97% Prec = 0.97 Recall = 0.99 F1 = 0.98
Elsadig et al. (2022)	12 URL character-based features	1	549,346	BERT and **CNN**	Acc = 96.66% Prec = 96.66% F1 = 96.63%
Liu et al. (2022)	URL and web page structure-based features	2	5,393	**CNN**	FPR = 0.0047 F1 = 0.9830 AUC = 0.9993
Alshingiti et al. (2023)	80 URL character-based features	1	20,000	**CNN**, LSTM and LSTM-CNN	Acc = 99.20% Prec = 99% Rec = 99.20% F1 = 99.20%
Our work	5 web page structure and contents, 14 URL, 1 WHOIS record, 2 TLS certificate, 4 web page reputation	5	26,115	LR, K-NN, DT, NB, SVM, ANN, RF, GB, LGBM, XGB, extra trees, **CatBoost**, FC-DNN, LSTM, 1D CNN	Acc = 99.02% FPR = 0.90% FNR = 1.03% Prec. = 0.99 Recall = 0.98 F1 = 0.99 AUC = 1.00

Note: The reported performances are of the best performing algorithms in bold.

6.1.1 Prediction performance

Our work has produced better performance across multiple metrics compared to most works. Note that Ma et al. (2009) evaluated their model against several datasets from different data sources with an accuracy ranging from 95% to 99% and error rates from 0.9% to 33.5%. Although their best performance is in the same range as ours, their least good is significantly worse than ours. The performance of our model against two independent datasets, as described in Sections 5.3.1 and 5.5, showed less variation than theirs, suggesting that our model is more robust. It can also be observed that works by Jain and Gupta (2018), Sahingoz et al. (2019), Elsadig et al. (2022), Liu et al. (2022) and Alshingiti et al. (2023) achieved closest performances to ours with the latter producing slightly better results than ours in some metrics. However, it should be noted that our work has introduced a set of new features. This merits our model by prolonging it from the exposure of possible detection evasion techniques as it takes time for phishers to learn about existing features before attempting to evade the features, thus the solution. Other key differences with our work are the use of a large number of different categories of features and the third-party features, of which their advantages were explained in Section 2.3. In addition, Sahingoz et al. (2019) and Elsadig et al. (2022) used datasets of multiples times the size of our dataset and still achieved similar performances with ours. As ML algorithms tend to increase their performances with an increase in the dataset sizes, enlarging our dataset to a size like theirs is likely to significantly boost our performance even further.

6.1.2 Diversity of features

Along with Ma et al. (2009), our model has used five different categories of both third party and local based features. Jain and Gupta (2018), Sahingoz et al. (2019) and Liu et al. (2022) used two categories of local based features, Xiang et al. (2011) and Shirazi et al. (2017) used four categories of third party and local based features whereas Elsadig et al. (2022) and Alshingiti et al. (2023) used only one category of local based features. Using a large number of feature categories increases the difficulty faced by an attacker in evading detection. Furthermore, local based features can easily be evaded by attackers than the features derived from the information obtained from third party services. For instance, a phishing web page created by copying a legitimate web page and slightly modifying URL is difficult to distinguish from a legitimate web page and is unlikely to be identified using the features proposed by Sahingoz et al. (2019). Third party features, which are the strongest predictors in our model, are always difficult to emulate or forge because the services are highly secured. We think that a mixture of third party and local based features is ideal for a more robust and evasion resistant solution.

6.1.3 Number of ML algorithms used for evaluation

While other works have used between 2- and 7-ML algorithms to evaluate their classifiers, our work has used 15. The advantage of exploring a large number of algorithms is that it allows us to draw a more concrete conclusion on the general effectiveness of the features for a given prediction problem. That is, a good range of performances for most algorithms in this case shows that the features have a minimal dependence on the learning capability of specific algorithms, thus, are robust for the problem.

6.1.4 Performance metrics

Our work has evaluated the classifier using eight performance metrics while the other works have used between 2 and 4 metrics. Important measures such as precision, recall and F1 score were not reported by some of the studies in the table whereas FNR was not reported explicitly by most studies. The latter is an important one as it measures the extent to which solutions misclassify true phishing websites, thus exposing end users to the attack. Evaluation using a small number of measures limits our understanding of the all-around effectiveness of the solutions.

6.2 Limitations of our model

Almost a half of our proposed best features were derived from third party services. Data retrieved from these services may be missing from time to time, for reasons including poor network connection, temporary unavailability of their servers or absent records in the databases. A high percentage of missing data is likely to reduce a detection performance. However, our experience during data collection suggests that scenarios which could give rise to missing data in third-party services are relatively rare in normal circumstances. In some features, for instance those related to TLS certificates and WHOIS records, missing data is quite common. In addition, we obtained better results in each model when we included all features with missing values under 50% compared to removing them completely. This suggests that our model, based on the training datasets used, can cope well with up to 50% of missing data in several features. This, however, might not be the case if training datasets with different data distributions are used.

6.3 Application of the proposed model

The web page loading time affects the web browsing user experience which in turn determines the percentage of users that are likely to decline to access the web page. The percentage increases as the loading time increases. As indicated in Table 13, our model takes 0.84 seconds to load a PDC web page on a desktop device and 6.31 seconds to predict it, giving a total prediction time of 7.2 seconds. According to MachMetrics (2021), at least 30% of users are likely to abandon the web page if the loading time exceeds 7 seconds. This is a significant loss to any website, especially the commercial ones. This means our model, if implemented as it is, will be less than ideal for real-time applications. However, the model’s prediction time can be reduced in two ways:

1 using more efficient python libraries and coding style

2 extracting most of the features in parallel.

For instance, all the third-party features can be extracted concurrently thus reducing their total time of 3.05 seconds to the longest time to extract one of their features, which is 1.19 seconds. Similarly, local based features can be extracted in parallel. The average time to extract one such feature is 0.14 seconds. The total time to load the web page and extract all the features in parallel would be 2.17 seconds (0.84 + 1.19 + 0.14). This time is less than 3 seconds, which is a range considered to be fast by many users according to MachMetrics (2021). With this improvement, our model can thus be implemented for real-time applications to protect users at the web browser as a built-in functionality or as a plug-in, for instance.

Alternatively, our model can be used to build a blacklist of phishing URLs by predicting phishing web pages from PDC web pages collected from various data sources such as emails and social media posts. The blacklist can then be used to defend users by integrating it with a web browser as a built-in functionality or as a plug-in. The blacklist can also be used to complement existing general-purpose blacklists for research purposes.

7 Conclusions

In this paper, we have proposed an ML based model that can instantly and accurately predict zero-day phishing PDC web pages using a novel set of highly diversified features. First, we investigated and proposed 35 features derived from various distinctive structural characteristics of phishing PDC web pages of which 26 features were found to be the most relevant features for the prediction task. The features were evaluated against 12 traditional ML and 3 DL algorithms whereby most of the algorithms were observed to produce relatively good performances. This indicates that our features are robust for this prediction problem. Of all the algorithms, CatBoost yielded the optimal prediction performance of *accuracy of 99.02%*, *FPR of 0.90%* and *FNR of 1.03%*. The 26 features are grouped into five categories namely web page structure and contents, URL structure, WHOIS records, TLS certificate and web page reputation. 9 of the features are based on third-party services while 17 of them were derived from the web page's structure. Our feature analysis indicated that novel and third-party based features are stronger predictors than the adopted and local based features. We also found that most of the features based on web page reputation against blacklisted phishing IP addresses and search engines are the most influential ones for the prediction whereas URL based features are among the least influential ones. The prediction time of the model was measured at 7.2 seconds but the time could go as low as 2.17 seconds if the features were to be extracted concurrently. The time suggests that the model can be used for real-time protection of users from accessing phishing PDC web pages without degrading their web browsing experiences. We also tested the model (without retraining) against a new dataset collected 14 months after collecting the first dataset. The results showed that the model performs consistently on different datasets, suggesting that the features are reliable for addressing the problem in long term. They also show that phishers do not vary their tactics in creating their websites frequently.

References

Acmetek (n.d.) 'What is the difference between domain validated (DV), organization validated and extended validation (EV) SSL?', [online] https://www.sslsupportdesk.com/what-is-the-difference-between-domain-validated-dv-organization-validated-and-extended-validation-ev-ssl/ (accessed June 2018).

Agarwal, N. (n.d.) *jQuery Get Value Of Input, Textarea and Radio Button* [online] https://www.formget.com/jquery-get-value-of-input/ (accessed November 2017).

Akamai (2019) *Phishing – Baiting the Hook* [online] https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-security-phishing-baiting-the-hook-report-2019.pdf (accessed May 2020).

Al-Garadi, M., Amr, M., Al-Ali, A., Du, X. and Guizani, M. (2018) *A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security*, arXiv preprint arXiv:1807.11023, DOI: 10.1109/COMST.2020.2988293, https://doi.org/10.1109/COMST.2020.2988293.

Alkhalil, Z., Hewage, C., Nawaf, L. and Khan, I. (2021) 'Phishing attacks: a recent comprehensive study and a new anatomy [review]', *Frontiers in Computer Science*, Vol. 3, No. 6, https://doi.org/10.3389/fcomp.2021.563060.

Allianz (n.d.) 'Cyber attacks on critical infrastructure' [online] https://www.agcs.allianz.com/news-and-insights/expert-risk-articles/cyber-attacks-on-critical-infrastructure.html (accessed March 2020).

Alshingiti, Z., Alaqel, R., Al-Muhtadi, J., Haq, Q.E.U., Saleem, K. and Faheem, M.H. (2023) 'A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN', *Electronics*, Vol. 12, No. 1, p.232, https://doi.org/https://doi.org/10.3390/electronics12010232.

Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A. and Marchetti, M. (2018) 'On the effectiveness of machine and deep learning for cyber security', *10th International Conference on Cyber Conflict (CyCon)*, IEEE, Estonia, 29 May–1 June, pp.371–390, https://doi.org/10.23919/CYCON.2018.8405026.

APWG (2016) *Phishing Activity Trends Report 4th Quarters 2016*, APWG [online] http://docs.apwg.org/reports/apwg_trends_report_q4_2016.pdf (accessed December 2016).

APWG (2023) *Phishing Activity Trends Report*, 4th Quarter 2022, APWG [online] https://docs.apwg.org/reports/apwg_trends_report_q4_2022.pdf?_gl=1*1uvke9m*_ga*MjAxNTY2NTkzMi4xNjkzMDYzMTIy*_ga_55RF0RHXSR*MTY5MzA2MzEyMS4xLjAuMTY5MzA2MzEyMS4wLjAuMA..&_ga=2.153107007.623434115.1693063122-2015665932.1693063122 (accessed August 2023).

AV-Comparatives (2016) *Anti-phishing Test* [online] https://www.av-comparatives.org/wp-content/uploads/2016/07/avc_phi_2016_en.pdf (accessed February 2019).

Ball, T. (2017) *Top 5 Critical Infrastructure Cyber Attacks* [online] https://www.cbronline.com/cybersecurity/top-5-infrastructure-hacks/ (accessed March 2020).

Barraclough, P., Hossain, M.A., Tahir, M., Sexton, G. and Aslam, N. (2013) 'Intelligent phishing detection and protection scheme for online transactions', *Expert Systems with Applications*, Vol. 40, No. 11, pp.4697–4706, https://doi.org/https://doi.org/10.1016/j.eswa.2013.02.009.

Berman, D., Buczak, A., Chavis, J. and Corbett, C. (2019) 'A survey of deep learning methods for cyber security', *Information*, Vol. 10, No. 4, p.122, https://doi.org/10.3390/info10040122.

Brattberg, E. and Maurer, T. (2018) *Russian Election Interference: Europe's Counter to Fake News and Cyber Attacks* [online] https://carnegieendowment.org/2018/05/23/russian-election-interference-europe-s-counter-to-fake-news-and-cyber-attacks-pub-76435 (accessed April 2020).

Broadley, C. (n.d.) *<input value=''>* [online] https://html.com/attributes/input-value/ (accessed November 2017).

Brownlee, J. (2014) *Classification Accuracy is Not Enough: More Performance Measures You Can Use* [online] https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/ (accessed August 2018).

Brownlee, J. (2016a) *A Gentle Introduction to XGBoost for Applied Machine Learning* [online] https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/ (accessed February 2020).

Brownlee, J. (2016b) *Logistic Regression for Machine Learning* [online] https://machinelearningmastery.com/logistic-regression-for-machine-learning/ (accessed March 2018).

Brownlee, J. (2016c) *Machine Learning Mastery with Python*, 14th ed. [online] https://machinelearningmastery.com/machine-learning-with-python/ (accessed April 2018).

Brownlee, J. (2016d) *Naive Bayes for Machine Learning* [online] https://machinelearningmastery.com/naive-bayes-for-machine-learning/ (accessed September 2018).

Brownlee, J. (2018) *How to Use ROC Curves and Precision-Recall Curves for Classification in Python* [online] https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/ (accessed January 2019).

Brownlee, J. (2020) *Introduction to Dimensionality Reduction for Machine Learning* [online] https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/ (accessed May 2021).

Brownlee, J. (2021a) *Gradient Boosting with Scikit-Learn, XGBoost, LightGBM, and CatBoost* [online] https://machinelearningmastery.com/gradient-boosting-with-scikit-learn-xgboost-lightgbm-and-catboost/ (accessed August 2022).

Brownlee, J. (2021b) *How to Develop an Extra Trees Ensemble with Python* [online] https://machinelearningmastery.com/extra-trees-ensemble-with-python/ (accessed August 2022).

Chauhan, N. (2020) *Decision Tree Algorithm, Explained* [online] https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html (accessed May, 2021).

Chen, K-T., Chen, J-Y., Huang, C-R. and Chen, C-S. (2009) 'Fighting phishing with discriminative keypoint features', *IEEE Internet Computing*, Vol. 13, No. 3, https://doi.org/10.1109/MIC.2009.59.

CNN (2020) *2016 Presidential Campaign Hacking Fast Facts* [online] https://edition.cnn.com/2016/12/26/us/2016-presidential-campaign-hacking-fast-facts/index.html (accessed April 2020).

Damiani, J. (2020) 'Google data reveals 350% surge in phishing websites during coronavirus pandemic', *Forbes* [online] https://www.forbes.com/sites/jessedamiani/2020/03/26/google-data-reveals-350-surge-in-phishing-websites-during-coronavirus-pandemic/?sh=7c4e66ff19d5 (accessed August 2023).

Dertat, A. (2017) *Applied Deep Learning – Part 4: Convolutional Neural Networks* [online] https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2 (accessed September 2019).

Dickson, B. (2019) 'What are artificial neural networks (ANN)?' [online] https://bdtechtalks.com/2019/08/05/what-is-artificial-neural-network-ann/ (accessed May 2021).

EC Council (2017) *Ethical Hacking and Countermeasures: Threats and Defence Mechanisms*, 2nd ed., EC-Council Press (accessed May 2018).

Elsadig, M., Ibrahim, A.O., Basheer, S., Alohali, M.A., Alshunaifi, S., Alqahtani, H. and Nagmeldin, W. (2022) 'Intelligent deep machine learning cyber phishing URL detection based on BERT features extraction', *Electronics*, Vol. 11, No. 22, p.3647, https://doi.org/https://doi.org/10.3390/electronics11223647.

ESET (2017) *ESET Anti-Phishing* [online] https://www.eset.com/us/anti-phishing/ (accessed August 2017).

Eubanks, N. (2012) *A Simple Guide to Using rel='alternate' hreflang='x'* [online] https://searchenginewatch.com/sew/how-to/2232347/a-simple-guide-to-using-rel-alternate-hreflang-x (accessed December 2017).

Federal Bureau of Investigations (FBI) (2018) *Business E-mail Compromise the 12 Billion Dollar Scam* [online] https://www.ic3.gov/media/2018/180712.aspx (accessed April 2020).

Federal Bureau of Investigations (FBI) (2021) *Internet Crime Report 2021* [online] https://www.ic3.gov/Media/PDF/AnnualReport/2021_IC3Report.pdf (accessed May 2021).

Gendre, A. (2015) *How Much Does A Spear Phishing Attack Cost?* [online] https://www.vadesecure.com/en/spear-phishing-cost/ (accessed April 2020).

Gendre, A. (2019) *4 Ways Hackers Use Phishing to Launch Ransomware Attacks* [online] https://www.vadesecure.com/en/3-ways-hackers-use-phishing-to-launch-ransomware-attacks/ (accessed March 2020).

Geotrust (n.d.) *Web Site Verification Search Site and Toolbar Fights Online Fraud and Phishing Scams* [online] https://www.trustico.co.uk/material/DS_TrustWatch.pdf (accessed February 2018).

Ghobril, S. (2015) 'What is href='#' and why is it used?', *Stack Overflow*, June 2018 [online] https://stackoverflow.com/questions/4855168/what-is-href-and-why-is-it-used (accessed June 2021).

Global Sign (n.d.) *What Are The Different Types of SSL Certificates?* [online] https://www.globalsign.com/en/ssl-information-center/types-of-ssl-certificate/ (accessed June 2018).

Google (2017) *Separate URLs* [online] https://developers.google.com/search/mobile-sites/mobile-seo/separate-urls (accessed December 2017).

Google (n.d.-a) *Google Safe Browsing* [online] https://safebrowsing.google.com/ (accessed March 2017).

Google (n.d.-b) *What is Safe Browsing?* [online] https://developers.google.com/safe-browsing/ (accessed March 2017).

Greenberg, A. (2017) *Everything We Know About Russia's Election-Hacking Playbook* [online] https://www.wired.com/story/russia-election-hacking-playbook/ (accessed April 2020).

Hancock, J.T. and Khoshgoftaar, T.M. (2020) 'CatBoost for big data: an interdisciplinary review', *Journal of Big Data*, Vol. 7, No. 1, p.94, https://doi.org/10.1186/s40537-020-00369-8.

Hara, M., Yamada, A. and Miyake, Y. (2009) 'Visual similarity-based phishing detection without victim site information', *Proc. IEEE Symposium on Computational Intelligence in Cyber Security*, IEEE, Nashville, TN, USA, 30 March–2 April, pp.30–36, https://doi.org/10.1109/CICYBS.2009.4925087.

IBM Security (2019) *Cost of a Data Breach Report 2019* [online] https://www.ibm.com/downloads/cas/ZBZLY7KL (accessed April 2020).

Internet Society (2016) *Global Internet Report 2016*, Internet Society [online] https://www.internetsociety.org/globalinternetreport/2016/wp-content/uploads/2016/11/ISOC_GIR_2016-v1.pdf (accessed April 2021).

Jain, A.K. and Gupta, B.B. (2018) 'Towards detection of phishing websites on client-side using machine learning based approach [journal article]', *Telecommunication Systems*, Vol. 68, No. 4, pp.687–700, https://doi.org/10.1007/s11235-017-0414-0.

Kaspersky (2015) *Secure Web Surfing With Kaspersky Lab Advanced Anti-Phishing Technology* [online] http://media.kaspersky.com/pdf/Kaspersky_Lab_Whitepaper_Anti_phishing.pdf (accessed August 2017).

Kavya (2020) *Types of SSL Certificates for a Secure Business Website* [online] https://serverguy.com/ssl/types-of-ssl-certificates/ (accessed April 2020).

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W. and Liu, T-Y. (2017) 'Lightgbm: a highly efficient gradient boosting decision tree', *Advances in Neural Information Processing Systems*, Vol. 30 [online] https://hal.science/hal-03953007/document (accessed August 2019).

Kent (2013) 'Bitdefender TrafficLight protects you from malware, phishing websites, and privacy trackers [Chrome, Firefox, Safari]' [online] https://dottech.org/132882/review-bitdefender-trafficlight/ (accessed March 2017).

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. and Inman, D.J. (2021) '1D convolutional neural networks and applications: a survey', *Mechanical Systems and Signal Processing*, Vol. 151, p.107398, https://doi.org/https://doi.org/10.1016/j.ymssp.2020.107398.

Koulopoulos, T. (2017) *60 Percent of Companies Fail in 6 Months Because of This (It's Not What You Think)* [online] https://www.inc.com/thomas-koulopoulos/the-biggest-risk-to-your-business-cant-be-eliminated-heres-how-you-can-survive-i.html (accessed April 2020).

Kumar, V. and Kumar, R. (2015) 'Detection of phishing attack using visual cryptography in ad hoc network', *2015 International Conference on Communications and Signal Processing (ICCSP)*, IEEE, pp.1021–1025, https://doi.org/https://10.1109/ICCSP.2015.7322654.

Kunal (2015) 'What should be the allowed percentage of missing values?', *Analytics Vidhya*, March 2021.

Kurtus, R. (2014) *Types of HTML Hyperlinks* [online] https://www.school-for-champions.com/web/html_hyperlinks.htm (accessed June 2018).

Lakshmi, V.S. and Vijaya, M. (2012) 'Efficient prediction of phishing websites using supervised learning algorithms', *Procedia Engineering*, Vol. 30, pp.798–805, https://doi.org/https://doi.org/10.1016/j.proeng.2012.01.930.

Lee, W. and Rotoloni, B. (2016) *Emerging Cyber Threats Report 2016*. Institute for Information Security & Privacy [online] http://www.iisp.gatech.edu/sites/default/files/documents/2016_georgiatech_cyberthreatsreport_onlinescroll.pdf (accessed July 2021).

Li, Y., Yang, Z., Chen, X., Yuan, H. and Liu, W. (2019) 'A stacking model using URL and HTML features for phishing webpage detection', *Future Generation Computer Systems*, Vol. 94, pp.27–39, https://doi.org/10.1016/j.future.2018.11.004.

Liu, D-J., Geng, G-G. and Zhang, X-C. (2022) 'Multi-scale semantic deep fusion models for phishing website detection', *Expert Systems with Applications*, Vol. 209, p.118305, https://doi.org/https://doi.org/10.1016/j.eswa.2022.118305.

Ma, J., Saul, L.K., Savage, S. and Voelker, G.M. (2009) 'Beyond blacklists: learning to detect malicious web sites from suspicious URLs', *Proc. 15th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, ACM, Paris, France, 28 June–1 July, pp.1245–1254, https://doi.org/https://doi.org/10.1145/1557019.1557153.

MachMetrics (2021) *Average Page Load Time in 2021* [online] https://machmetrics.com/speed-blog/average-page-load-time-in-2021/ (accessed July 2021).

Madley-Dowd, P., Hughes, R., Tilling, K. and Heron, J. (2019) 'The proportion of missing data should not be used to guide decisions on multiple imputation', *Journal of Clinical Epidemiology*, Vol. 110, pp.63–73, https://doi.org/https://doi.org/10.1016/j.jclinepi.2019.02.016.

Medvet, E., Kirda, E. and Kruegel, C. (2008) 'Visual-similarity-based phishing detection', *Proc. 4th International Conference on Security and Privacy in Communication Networks*, ACM, Istanbul, Turkey, 22–25 September, p.22, https://doi.org/https://doi.org/10.1145/1460877.1460905.

Meier, C. (2014) *A Beginners Introduction to the Canonical Tag* [online] https://unamo.com/blog/general/beginners-introduction-canonical-tag (accessed December 2017).

Microformats (2016) *Rel-Alternate* [online] http://microformats.org/wiki/rel-alternate (accessed December 2017).

Moolayil, J. (2019) *Learn Keras for Deep Neural Networks*, 1st ed., Apress, California, US, https://doi.org/10.1007/978-1-4842-4240-7.

Müller, A. and Guido, S. (2017) *Introduction to Machine Learning with Python*, 1st ed., O'Reilly Media, California, US.

Navlani, A. (2019) *Support Vector Machines with Scikit-learn* [online] https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python (accessed May 2021).

Nguyen, M. (2018) *Illustrated Guide to LSTM's and GRU's: A Step by Step Explanation* [online] https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 (accessed September 2019).

Nicholson, C. (n.d.) *A Beginner's Guide to Neural Networks and Deep Learning* [online] https://wiki.pathmind.com/neural-network (accessed May 2021).

Omg (2009) 'What does 'javascript:void(0)' mean?', *Stack Overflow* [online] https://stackoverflow.com/questions/1291942/what-does-javascriptvoid0-mean (accessed August 2021).

Pathak, M. (2018) *Handling Categorical Data in Python* [online] https://www.datacamp.com/community/tutorials/categorical-data (accessed August 2021).

PCHelp (2002) 'How to obscure any URL', *PCHelp* [online] http://www.pc-help.org/obscure.htm (accessed December 2017).

Phi, M. (2018) *Illustrated Guide to LSTM's and GRU's: A Step by Step Explanation* [online] https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 (accessed March 2021).

PhishLabs (2020) *2019 Phishing Trends and Intelligence Report* [online] https://info.phishlabs.com/hubfs/2019%20PTI%20Report/2019%20Phishing%20Trends%20and%20Intelligence%20Report.pdf (accessed April 2020).

Pompon, R. (2019) *Three Ways to Hack the U.S. Election* [online] https://www.f5.com/labs/articles/threat-intelligence/three-ways-to-hack-the-u-s--election (accessed April 2020).

Ponemon Institute (2015) *The Cost of Phishing and Value of Employee Training*, Ponemon Institute [online] https://info.wombatsecurity.com/hubfs/Ponemon_Institute_Cost_of_Phishing.pdf (accessed April 2021).

Ray, S. (2017) 'Understanding support vector machine (SVM) algorithm from examples (along with code)' [online] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ (accessed May 2021).

Retruster (n.d.) *The True Cost of a Phishing Attack* [online] https://retruster.com/blog/phishing-attack-true-cost.html (accessed April 2020).

Robertckl (2014) 'Types of SSL certificates – choose the right one', *Symantec Official Blog* [online] https://www.symantec.com/connect/blogs/types-ssl-certificates-choose-right-one (accessed June 2021).

Rodríguez, J. (2019) 'Most common attack vector over critical infrastructures' [online] https://www.cipsec.eu/content/most-common-attack-vector-over-critical-infrastructures (accessed March 2020).

Sahingoz, O.K., Buber, E., Demir, O. and Diri, B. (2019) 'Machine learning based phishing detection from URLs', *Expert Systems with Applications*, Vol. 117, pp.345–357, https://doi.org/https://doi.org/10.1016/j.eswa.2018.09.029.

SecureWorks (2019) *COBALT DICKENS Goes Back to School…Again* [online] https://www.secureworks.com/blog/cobalt-dickens-goes-back-to-school-again (accessed August 2021).

Shirazi H., Haefner, K. and Ray, I. (2017) 'Fresh-Phish: a framework for auto-detection of phishing websites', *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, San Diego, CA, USA, 4–6 August.

Shwartz-Ziv, R. and Armon, A. (2022) 'Tabular data: deep learning is not all you need', *Information Fusion*, Vol. 81, pp.84–90, https://doi.org/https://doi.org/10.1016/j.inffus.2021.11.011.

Sillipo, R. and Widmann, M. (2019) *3 New Techniques for Data-Dimensionality Reduction in Machine Learning* [online] https://thenewstack.io/3-new-techniques-for-data-dimensionality-reduction-in-machine-learning/ (accessed May 2021).

Somogyi, S. and Miller, A. (2017) *Safe Browsing: Protecting more than 3 Billion Devices Worldwide, Automatically* [online] https://www.blog.google/technology/safety-security/safe-browsing-protecting-more-3-billion-devices-worldwide-automatically/ (accessed May 2020).

Sophos (2019) *Don't Take the Bait* [online] https://secure2.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/Dont-Take-The-Bait.pdf (accessed June 2020).

Trend Micro (n.d.) *Trend Micro: Phishing* [online] http://www.antivirus.co.uk/Internet-Security-Info/Phishing/ (accessed May 2019).

Universal Class (n.d.-a) *How to Script Forms with JavaScript* [online] https://www.universalclass.com/articles/computers/javascript/how-to-script-forms-with-javascript.htm (accessed November 2017).

Universal Class (n.d.-b) *User Input and Output in JavaScript* [online] https://www.universalclass.com/articles/computers/javascript/user-input-and-output-in-javascript.htm (accessed November 2017).

Valk, J. (2016) *re=canonical: The Ultimate Guide* [online] https://yoast.com/rel-canonical/#when-to-canonicalize (accessed November 2017).

VanderPlas, J. (2017) *Python Data Science Handbook*, 1st ed., O'Reilly Media (accessed May 2019).

Verizon (2018) *2018 Data Breach Investigations Report* [online] https://enterprise.verizon.com/resources/reports/DBIR_2018_Report_execsummary.pdf (accessed May 2020).

Verma, S. (2019) *Understanding 1D and 3D Convolution Neural Network | Keras* [online] https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610 (accessed March 2020).

W3Schools (n.d.-a) *HTML <label> Tag* [online] https://www.w3schools.com/tags/tag_label.asp (accessed November 2017).

W3Schools (n.d.-b) *HTML Input Types* [online] https://www.w3schools.com/html/html_form_input_types.asp (accessed November 2017).

W3Schools (n.d.-c) *JavaScript Popup Boxes* [online] https://www.w3schools.com/js/js_popup.asp (accessed November 2017).

Warburton, D. and Pompon, R. (2019) *2019 Phishing and Fraud Report* [online] https://www.f5.com/labs/articles/threat-intelligence/2019-phishing-and-fraud-report (accessed April 2020).

Webroot (2019) *The 2019 Webroot Threat Report* [online] https://www-cdn.webroot.com/9315/5113/6179/2019_Webroot_Threat_Report_US_Online.pdf (accessed May 2020).

Wenyin, L., Liu, G., Qiu, B. and Quan, X. (2012) 'Antiphishing through phishing target discovery', *IEEE Internet Computing*, Vol. 16, No. 2, pp.52–61, https://doi.org/https://doi.org/10.1109/MIC.2011.103.

Worcester, P. (2019) *Comparison of Grid Search and Randomized Search Using Scikit Learn* [online] https://blog.usejournal.com/a-comparison-of-grid-search-and-randomized-search-using-scikit-learn-29823179bc85 (accessed May 2021).

Xiang, G., Hong, J., Rose, C.P. and Cranor, L. (2011) 'Cantina+: a feature-rich machine learning framework for detecting phishing web sites', *ACM Transactions on Information and System Security (TISSEC)*, Vol. 14, No. 2, p.21, https://doi.org/https://doi.org/10.1145/2019599.2019606.

Yiu, T. (2019) *Understanding Random Forest* [online] https://towardsdatascience.com/understanding-random-forest-58381e0602d2 (accessed February 2020).

Zhao, R., John, S., Karas, S., Bussell, C., Roberts, J., Six, D. and Yue, C. (2016) 'The highly insidious extreme phishing attacks', *Proc. 25th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, Waikoloa, HI, USA, 1–4 August, pp.1–10, https://doi.org/10.1109/ICCCN.2016.7568582.