Journal of Ambient Intelligence and Humanized Computing (2023) 14:9233–9251
https://doi.org/10.1007/s12652-022-04426-3

ORIGINAL RESEARCH

Highly accurate phishing URL detection based on machine learning

Sajjad Jalil[1] · **Muhammad Usman**[1] · **Alvis Fong**[2]

Received: 6 August 2021 / Accepted: 14 September 2022 / Published online: 8 October 2022

Abstract
Phishing is a persistent and major threat on the internet that is growing steadily and dangerously. It is a type of cyber-attack, in which phisher mimics a legitimate website page to harvest victim's sensitive information, such as usernames, emails, passwords and bank or credit card details. To prevent such attacks, several phishing detection techniques have been proposed such as AI based, 3rd party, heuristic and content based. However, these approaches suffer from a number of limitations that needs to be addressed in order to detect phishing URLs. Firstly, features extracted in the past are extensive, with a limitation that it takes a considerable amount of time to extract such features. Secondly, several approaches selected important features using statistical methods, while some propose their own features. Although both methods have been implemented successfully in various approaches, however, these methods produce incorrect results without amplification of domain knowledge. Thirdly, most of the literature has used pre-classified and smaller datasets, which fail to produce exact efficiency and precision on large and real world datasets. Fourthly, the previous proposed approaches lack in advanced evaluation measures. Hence, in this paper, effective machine learning framework is proposed, which predicts phishing URLs without visiting the webpage nor utilizing any 3rd party services. The proposed technique is based on URL and uses full URL, protocol scheme, hostname, path area of the URL, entropy feature, suspicious words and brand name matching using TF-IDF technique for the classification of phishing URLs. The experiments are carried out on six different datasets using eight different machine learning classifiers, in which Random Forest achieved a significant higher accuracy than other classifiers on all the datasets. The proposed framework with only 30 features achieved a higher accuracy of 96.25% and 94.65% on the Kaggle datasets. The comparative results show that the proposed model achieved an accuracy of 92.2%, 91.63%, 94.80, 96.85% on benchmark datasets, which is higher than the existing approaches.

Keywords Phishing detection · URL detection · Cybercrime · Machine learning · Random forest (RF) · Decision tree (J48)

1 Introduction

Phishing is a cybercrime attack that is increasing steadily and dangerously. Phishing attack could affect any type of business, whether it is governmental, social, financial or individual. It is a type of attack that occurs, when someone creates a malicious or duplicate webpage of the legitimate website. Typically, phisher trap victims into their cage by sending phishing links through different channels, involving social networks, emails, text messages and other communication medians as a tool to commit stealth of individual's personal data such as email, passwords, phone numbers and credit card details. Typically, phisher misleads user by developing forms pursuing towards a deceptive trap to sign-in or to make the user put-in their real detail and allocate the chance of stealth of personal data to the phisher. In order to carry out phishing attacks, attackers may use emails, web pages and malwares.

Figure 1 illustrate a comparison of phishing and legitimate webpage. In this figure, phisher created duplicate webpage of PayPal login which appears to be the exact similar copy of the original webpage and host it in a domain that is free or charged. Differentiating between genuine and

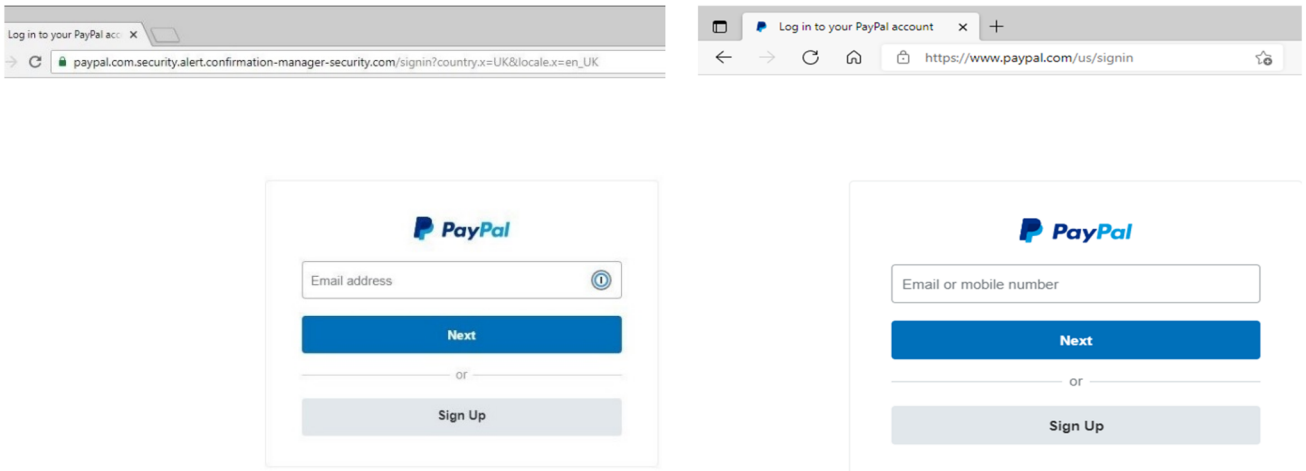
✉ Alvis Fong
alvis.fong@wmich.edu

Sajjad Jalil
sajjadjalil93@gmail.com

Muhammad Usman
dr.usman@szabist-isb.edu.pk

[1] Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology University, Islamabad, Pakistan

[2] Western Michigan University, Kalamazoo, USA



(a) Phishing webpage mimics PayPal login Page

(b) Legitimate PayPal login page

Fig. 1 Examples of Phishing and Legitimate PayPal Login page

phishing web pages, when they appear alike is difficult for human eyes. Sensitive information can be stolen if users fill out the form and in this case, email and password could be stolen by the phisher. However, the only thing that differs these webpages from each other is the URL. Thus, URL can be distinguishable and easy to detect any phishing URLs successfully.

According to Webroot threat report (2020), phishing URLs significantly grew by 640% throughout the year and the most prominent way to fool users was the use of Tiny-Urls. Most of the phishing links disappear in an hour and more than 40% of the URLs were found with HTTPS.

Similarly, Anti-Phishing Working Group (APWG 2020) issued a report in 2020, which states that 571,764 unique phishing URLs were detected from July to September, 2020. Moreover, the most affected sector was SAAS/Webmail that is targeted by the phishers.

Furthermore, these threats have drastically decreased consumers' interest in the use of commercial websites. For further estimation, the last 8 years data has been gathered from APWG (2020), which states that phishing webpages are rapidly increasing each year, which put the internet world as a dangerous and unsecured platform. Figure 2 illustrates that the year 2020 was the most prominent year for the phishers. This information could be unfolded after having detected about a half of 1 million phishing links during the run of said year.

Some of the existing work for phishing detection also includes review papers and surveys such as Alsharnouby et al. (2015), Chiew et al. (2018), Dou et al. (2017), Jalil and Usman (2020) which aim to compare the existing techniques. The survey in Alsharnouby et al. (2015) reveals that 47% individuals cannot identify the phishing URLs and the users are easily deceived by the phisher. Similarly, phishing attacks were concentrated in depth by Chiew et al. (2018) through their medium, vector characteristics and advanced methodologies. It is not necessary to focus only on client guidance as a protective measure in a phishing attack. Their analysis indicate of the importance to enhance clever mechanisms to fight these advanced methodologies, as such counter measures would have the choice of detecting and disabling both current and new phishing attacks. Moreover, Dou et al. (2017) has divided the phishing attack into 5 steps named; reconnaissance, weaponization, distribution, exploitation and exfiltration. The process of phishing is discussed in detail from selecting the target website to extracting the information of the victims in their survey. The existing approaches are critically evaluated against each other in terms of techniques, datasets, used features and performance measures in Jalil and Usman (2020).

There exists massive research work for phishing detection and in each approach the main focus is on phishing webpage and phishing email detection. In this paper, our focus is on the webpage (URL) based approach and we have compared our approach with those techniques that have classified phishing URLs or webpages. Furthermore, just like phishing have many characteristics, so does phishing detection methods and techniques. There are two types of detection methods, which can be classified into following categories:

Third party-oriented detection Most of the research work is dependent on the 3rd party based approaches such as blacklist and whitelist. The latest browsers, such as Google Chrome, Microsoft Edge, Opera and Mozilla, keep a list of prohibited and allowed URLs. The blacklist repository contains phishing URLs, while the whitelist repository contains legitimate URLs. Even reputable sites that aren't listed in whitelist repository can also be blocked from the access of the browser. Both blacklist and whitelist based approaches

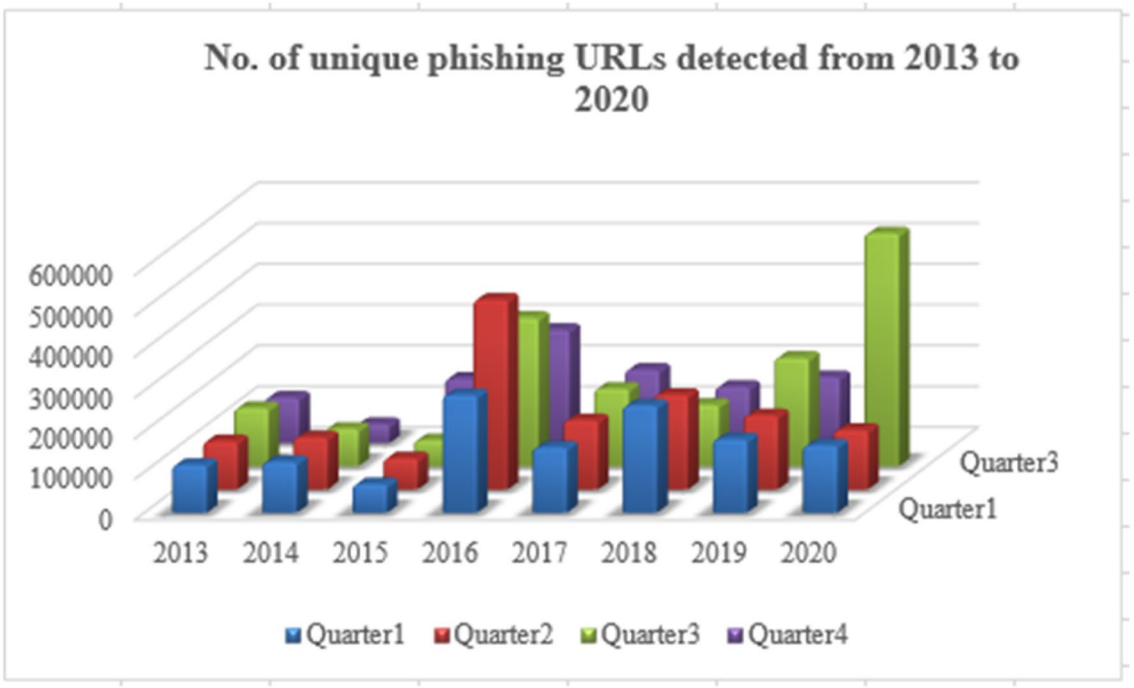


Fig. 2 No. of unique phishing URLs detected by APWG (2020) from 2013 to 2020

fail, when zero-day phishing sites (phishing sites that are less than a day old) are encountered. That's possible to get around it simply changing the URL slightly.

Heuristic oriented detection This is the most famous method that performs very well in terms of efficiency, precision and accuracy. This method is further divided into 3 types; visual similarity, content similarity and URL based techniques. Visual similarity based techniques detects phishing web-pages by comparing the web-pages visually by applying the technique of digital image processing. On the other hand, content similarity based techniques compare the contents of the web-page such as code and DOM contents. The limitation of these approaches are seldom controllable, as the phisher might easily overcome this security mechanism by changing a small portion of the web-page without even changing the contents of the web-page. Lastly, the URL based method is only dependent on the URL characteristics and is further divided into two parts; Lexical and Host based. Lexical based features are extracted from the URL characteristics and strings, while Host based features are extracted from WHOIS database such as search engine indexing, page rank and WHOIS record. Amongst all these techniques, URL (Lexical based) is the only technique that is not web dependent, its processing time is very low and the accuracy of phishing URL detection is very high. Table 1 illustrates the comparison of these techniques.

The objective of this paper is to develop a most suitable phishing URL detection framework, which helps to identify phishing URLs successfully. For achieving the objective, several phishing detection models have been proposed such as AI based, 3rd party, heuristic and content based. However, these approaches suffer from a number of limitations that needs to be addressed in order to detect phishing URLs. Firstly, feature engineering is essential step in any phishing

Table 1 Comparison of Anti-phishing detection techniques

Detection techniques	Web access?	Zero-hour detection?	Processing time	Accuracy
Blacklist	Yes	No	High	Low
Whitelist	Yes	No	High	Low
Visual similarity	Yes	No	High	High
Content similarity	Yes	No	High	High
URL(Host)	Yes	Yes	Mid	High
URL (Lexical)	No	Yes	Low	High

detection method, as the performance of the method critically depends on it. Although features extracted in the past are extensive with a limitation that it takes a considerable amount of time to extract such features. Similarly, using features based on third party produce high false positive rates. Secondly, several approaches selected important features using statistical methods while some approaches proposed their own features. Although both methods have been implemented successfully in various approaches, however, these methods produce incorrect results without amplification of domain knowledge. Thirdly, most of the literature has used pre-classified and smaller datasets for classification and evaluation of proposed models, which fails to produce the exact efficiency and precision on large and real world datasets. Fourthly, previously proposed approaches lack in advanced evaluation measures that can determine the performance of the particular model for phishing detection.

The above mentioned issues have motivated us to present accurate framework for phishing URL detection that can classify phishing URLs effectively and accurately. In this paper, we proposed phishing URL detection framework using lexical based features, depending neither on any third party services nor needs to visit the webpage for feature extraction. The features are extracted directly from the URL string. From out of 100 features, top 30 features are selected using domain knowledge and ReliefF technique. The proposed framework is experimented on six datasets using 8 different machine learning classifiers. Moreover, the prediction of phishing emails are not covered in this paper. In addition, the experiments are only observed using well-known classification algorithms.

Following are the advantages and contributions of the proposed framework:

Advantages of the proposed framework

- *Not utilizing 3rd party based features* The proposed framework does not extract features that are based on third party and not utilizing the internet to extract features.
- *Less time to extract features* The proposed framework extracts features directly from the URL, means only the lexical based features are extracted. Thus, the process of feature extraction is fast and it doesn't require visiting the webpage to extract such features.
- *Secure of drive by download* As the proposed framework extracts features from URL itself, thus visiting the webpage is prohibited and user is safe from downloading the malwares in their systems from the phishing webpage.
- *Language independent* Since, features are extracted from the URL, so the proposed framework can detect phishing web-pages written in any language.

Contributions of the proposed framework

- We have proposed a feature rich framework, initially extracted 100 features from the URL and selected top 30 features using domain knowledge and ReliefF technique.
- We have proposed 10 novel features (F2–F4, F7–F8, F12, H4, P2, P5, P11) along with 20 existing features adopted from the literature.
- This paper has evaluated six datasets using eight different machine learning classifiers.
- We have proposed a novel feature named as brand matching using Term Frequency-Inverse Document Frequency (TF-IDF) for classification of phishing URLs.
- We have proposed Suspicious words for phishing URL detection.

The rest of the paper is organized as follows. In Sect. 2, literature is reviewed. In Sect. 3, proposed framework and methodology is discussed. In Sect. 4, experiments and results are discussed. The discussion and limitations of the proposed framework is presented in Sect. 5. Finally, the proposed model is concluded in Sect. 6.

2 Literature review

This section provides an overview of the past work in the field of phishing URL detection. As we mentioned earlier, phishing detection can be categorize into two categories. First, third party based detection which matches phishing links in blacklist and whitelist repositories. The limitation of such approaches is that they are unable to detect newly launch websites. Secondly, heuristic based approaches are such approaches which uses machine learning, visual and content similarity based techniques to detect phishing websites. Visual similarity based approaches compare the webpages visually and content based approaches compare the code and its contents. The limitation of these approaches are seldom controllable, as the phisher might easily overcome this security mechanism by changing a small portion of the web-page without even changing the contents of the page. While machine learning based technique trains a particular classifier with extracted features and output the webpage as phishing or legitimate. Therefore, machine learning (ML) approaches are more effective, accurate and gained popularity in terms of phishing detection. Further, in this section, some of the ML based techniques are overviewed.

ML approaches are based on feature engineering and Rao et al. (2019) proposed 3 type of feature engineering techniques such as TF-IDF, human crafted and combine them to get better results. 47 features are extracted from the given URL and used 3 different datasets from different sources such as PhishTank (2022), Alexa (2022) and DMOZ. The

datasets are divided randomly using 10-Fold CV. All the experiments are performed using python and compared six ML classifiers in which Random Forest (RF) achieved the highest accuracy of 94.32% on dataset1 using 35 features.

Similarly, the same datasets is utilized by Korkmaz et al. (2020) using 8 different classifiers and extracted 48 best features. The efficiency of phishing URL detection is improved by using RF and achieved the accuracy of 94.59%. Moreover, Sahingoz et al. (2019) proposed real-time system using the dataset that consists of total 73,575 URLs comprises of phishing and legitimate URLs. For feature extraction, Natural Language Processing (NLP) method was used and extracted features such as brand name or number of characters from the URL. Once the features are extracted, the final dataset was trained and tested on WEKA tool. RF achieved the accuracy of 97.98% using NLP features.

Moreover, the techniques (Jeeva and Rajsingh 2016; Banik and Sarma 2018) have identified phishing URLs by classifying them with lexical features such as length of URL, number of dots, special characters, digits, subdomains. Both techniques have extracted less number of features using smaller datasets and achieved better accuracy for phishing URL detection.

Furthermore, the techniques (Jagadeesan et al. 2018; Pandey et al. 2019; Kulkarni and Brown 2019; Hutchinson et al. 2018; Feng et al. 2018; Zhu et al. 2019; Shahrivari et al. 2020; Aburub and Hadi 2021) have utilized pre-classified datasets from UCI repository which has less number of attributes and the datasets are very old. The datasets have different number of extracted features that further needs selection for better performance of the model. Jagadeesan et al. (2018) have utilized a dataset of 2456 instances and achieved 95.11% accuracy using RF. Whereas, Pandey et al. (2019), Kulkarni and Brown (2019) have utilizes the dataset of 1353 instances and achieved 94.74% and 91.5% accuracies using Hybrid RF-SVM and Decision Tree (DT). Moreover, the remaining approaches have utilized dataset of 11,055 instances in which Hutchinson et al. (2018) achieved 96.5% accuracy using RF and Shahrivari et al. (2020) achieved 98.32% accuracy using XGBoost. Feng et al. (2018) achieved 97.71% accuracy using Neural Network that is composed of three layers. Similarly, Zhu et al. (2019) achieved 96.44% accuracy using Neural Network and selected optimal features using Feature Validity Value (FVV). Aburub and Hadi (2021) achieved 83.8% accuracy using Association Classification (AC) algorithm.

The approaches (Abuzuraiq et al. 2020; Joshi and Pattanshetti 2019; Chiew et al. 2019; Tan et al. 2016) use the pre-classified dataset of 10,000 instances from Mendeley repository and uses lexical, host and 3rd party based features which is not efficient for real time systems. The tool that has been utilized in these techniques are WEKA and RF is used for classification of the phishing URLs. These techniques have only calculated accuracy for the prediction of phishing URLs and lack in other advanced evaluation measures. Tan et al. (2016) proposed a technique named PhishWho which is based on four components: extraction of identity keywords, look for URLs in search engine, find the target domain name and matching of three-tier identities. After experimentation, the approach achieved 96.10% accuracy.

Jain and Gupta (2018a, 2018b) proposed two different approaches using different datasets. In the first approach, the dataset of 35,451 URLs are collected from PhishTank and DMOZ repository and extracted 14 features from it. SVM achieved 91.28% accuracy using train test ratio of 40:60. Similarly, in second approach, the dataset of 2544 URLs are collected from PhishTank and Alexa and extracted 12 content based features from it. WEKA tool is used for experimentation in which Logistic Regression (LR) achieved 98.42% accuracy using train test ratio of 90:10. In both approaches, author has obtained the highest accuracy on smaller dataset which is not efficient for real time systems and they have utilized 3rd party features, which takes more time while training the model.

Rao and Pais (2018) proposed hybrid ML approach by using 16 different type of features such as hyperlink based, 3rd party and URL based features to detect phishing URLs. Li et al. (2019) proposed real time stacking model by combining ML algorithms GBDT, XGBoost and LightGBM to detect phishing URLs. The approach extracted 20 features from both URL and content of the website. The technique uses two layers model in which features are generated using GBDT, XGBoost and LightGBM in the first layer and then fed these features into the second layer to obtain new features. After the experimentation, the approach able to achieved 97.30% accuracy. Chavan et al. (2019) extracted 19 features from the dataset and achieved 96.82% accuracy using DT on smaller dataset of 1782 URLs that is collected from Kaggle repository.

Gupta et al. (2021) proposed a lightweight phishing detection technique on a dataset of ISCXURL-2016 having 11,964 instances. The technique extract nine lexical based features from the URL and achieved 99.57% accuracy using Random forest classifier.

Sadique et al. (2020) proposed a two-phase model. In the first phase, features were extracted from the URL, such as GeoIP, WHOIS, and lexical. If confidence level is too low, then features are labelled by human. The approach uses drop-column technique to select top 20 important features from 142 extracted features. Li and Wang (2017) proposes a model named as PhishBox; solution to verify and detect phishing websites. The ensemble model is intended to analyze the phish data in the first step, and active learning is introduced to minimize the expense of manual labelling. The validated phishing data is used in the second stage to train the detection model. The content features

of the URL and website are fed to an ensemble learning module as input. The approach achieved 95% accuracy using RF ensemble approach.

El Aassal et al. (2020) proposes ready to use framework called PhishBench, which helps to evaluate and analyze the available detection features and to thoroughly understand test conditions such as model requirements, datasets used, classifier performance and output measurements.

HTMLPhish (Opara et al. 2020) was an attempt to automate feature extraction from HTML pages using Convolutional Neural Network (CNN). It obtained 97.2% detection accuracy by using HTML pages. Bahnsen et al. (2017) have presented a high-precision LSTM network-based approach. There was no need for manual feature extraction because the approach has used URLs only. After an encoding step, the URLs are supplied to the LSTM network, which reduced the detection time. The LSTM is utilized for the first time in phishing detection, and it outperformed with 98.7% accuracy.

Chatterjee and Namin (2019) proposes a model to take deeper look into the clustering problem by exploring hidden features using deep reinforcement learning. The model adapts the behavior of phishing webpage dynamically and learn the features by itself. The approach is able to obtain 90.1% accuracy using the dataset of 73,575 URLs. Al-Alyan and Al-Ahmadi (2020) achieved 95.78% accuracy using CNN model.

Yang et al. (2019) proposes multidimensional feature selection approach using deep learning for phishing detection. 24 hybrid features are extracted from the URL. The CNN method is used to extract features that are correlated and LSTM (long short term memory) is used to record dependent features and context semantic from the URL characters. The hybrid approach of CNN-LSTM have achieved 98.09% accuracy using much larger dataset.

At last, a brief review is presented in terms of limitations and different parameters to identify gaps in current literature. The review include dataset size, feature engineering techniques, classifiers and models used for phishing URL detection.

Firstly, feature engineering is essential step in any phishing detection method, as the performance of the method critically depends on it. Although features extracted in past work are extensive with a limitation that it takes a considerable amount of time to extract such features. Similarly, using features such as third party based produce high false positive rates. Moreover, several approaches selected important features using statistical methods, while some approaches proposed their own features. Although both methods have been implemented successfully in various approaches, however, these methods produce incorrect results without amplification of domain knowledge.

Secondly, the existing techniques have been tested mainly with benchmark datasets and pre-classified datasets as mentioned in literature. The pre-classified datasets are collected from UCI repository (UCI 2022) which has old datasets that does not provide a true representation of the existing phishing datasets. Moreover, most of the datasets in machine learning literature are smaller in size. In the literature, most of the features that has been extracted from URL and webpage is based on 3rd party and host, which takes considerable amount of time to extract such features. The dataset directly adds the impact on the performance of the model.

Thirdly, previously proposed approaches lack in advanced evaluation measures that can determine the performance of the particular model for phishing detection. Most of the approaches have focused only on the accuracy while TPR and FPR show the true efficiency of the model. So, there is a strong need of a framework that can not only increase the true positive rate but also decrease the false positive rate for phishing URL detection.

3 Methodology for phishing URL detection

The general purpose of phishers is to get sensitive data from individual by replicating legitimate webpage, so they can trap user into their cage. Various URL obscuring strategies are used by criminals to trick consumers to expose sensitive details that can be misused.

The main focus of this paper is to detect phishing URLs instantly by using only URL characteristics. This can be done by using the URL string only, without visiting the website. This section provide an overview of our proposed model; the datasets used, preprocessing steps, feature engineering, experimental setup and performance measures.

The architecture of the proposed framework is shown in Fig. 3. The framework is divided into 3 phases: Firstly, the collected dataset is preprocessed by removing missing and duplicate URLs, extract features from each of the URL of the dataset, select top best 30 features out of 100 extracted features and then normalize the feature form dataset. Secondly, the dataset is divided into 70% for training and 30% for testing the model. The training dataset is to train the machine learning model and then evaluate the performance of the trained model on testing dataset. Thirdly, the model is evaluated using different evaluation measures on testing dataset and output the best classifier that has achieved highest accuracy and efficiency in terms of detecting phishing URLs.

3.1 Dataset & URL characteristics

A concise overview of the datasets and URL components are discussed in this section. URL implies a standardized

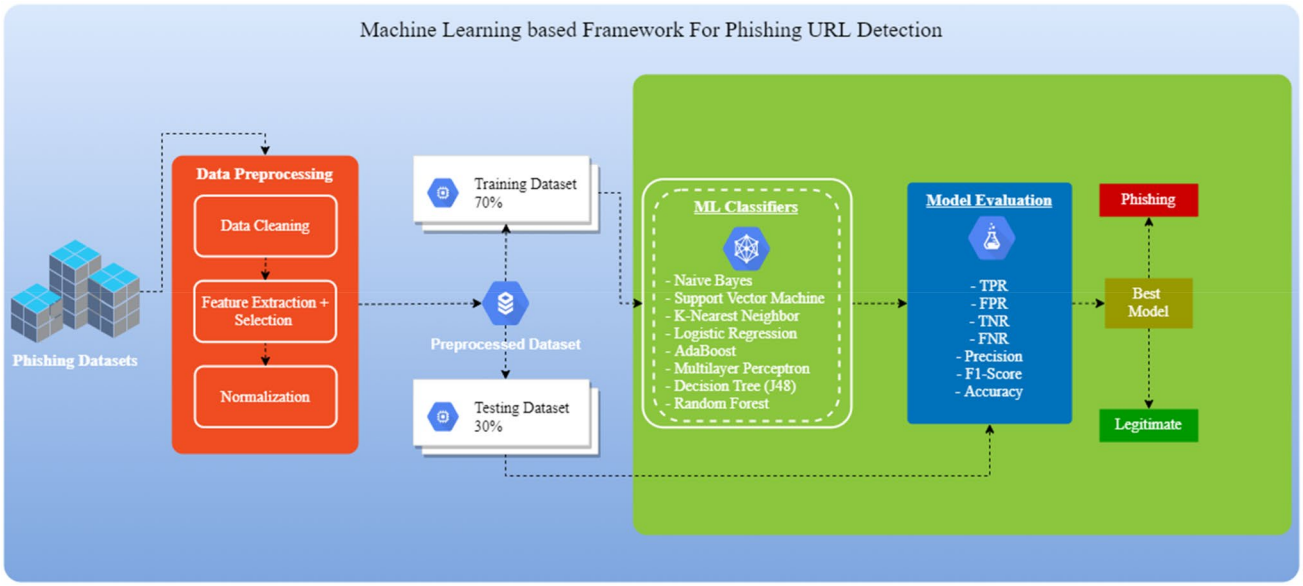


Fig. 3 Phishing URL detection framework

locator of resource. URL is used to search a resource such as content pages, images, files and more. The typical structure and different URL components with an example are shown in Fig. 4. The URL usually composed of following elements:

- *Scheme* protocol of the URL such as http or https.
- *Host* Specify the levels of domains such as primary or main domain, Sub-domain and top level domain (TLD).
- *Path* The address of the specific resource.
- *Query* The values and strings that comes up after "?".
- *Fragment* The direction to the sub section in a page and it's preceded by "#".

Phisher usually hides the hostname and domain name of the URL; to fool user that the link belongs to the legitimate website. Phisher usually use long URLs to hide the true identity of the website, so end-user cannot recognize the illegitimacy of the website. Phisher can register any new domain name and then add sub-domains to that primary domain. The sub-domain and path section of the URL is fully controlled by the phisher and phisher can add anything in it. Most of the phishing URLs have long sub-domains, so they hide the primary domain name to trap user into their cage.

Figure 5 shows the process of dataset transformation into feature set. We have collected the datasets in two column CSV format, in which first column has URLs and the second column indicates a class variables. 0 indicates that the URL is legitimate and 1 indicates that the URL is phish. We have written a Python script to extract features from each URL of the dataset. Once all the URLs of the dataset are parsed into the script, the new file having all the features are generated and saved into the system. The process of URL transformation and feature extraction is useful for machine learning models. Moreover, we have considered that all the URLs are correctly labelled whether the given URL is phish or legitimate in the datasets.

In this paper, we have collected six datasets from three different sources. Two bigger datasets are collected from Kaggle repository whereas, three datasets are collected from

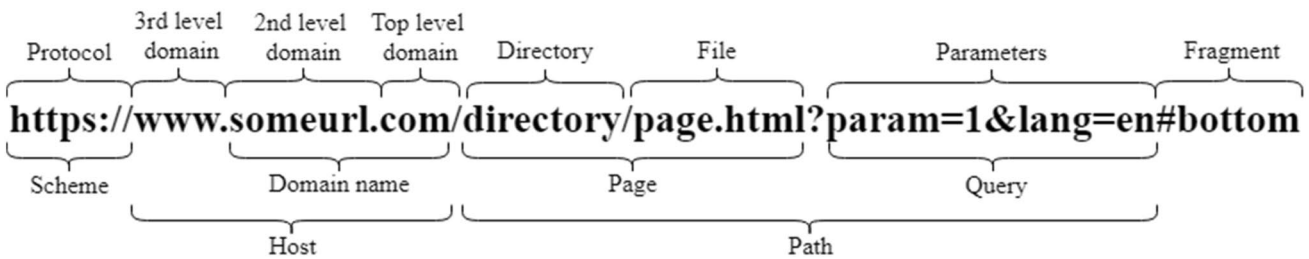


Fig. 4 Typical representation of the URL

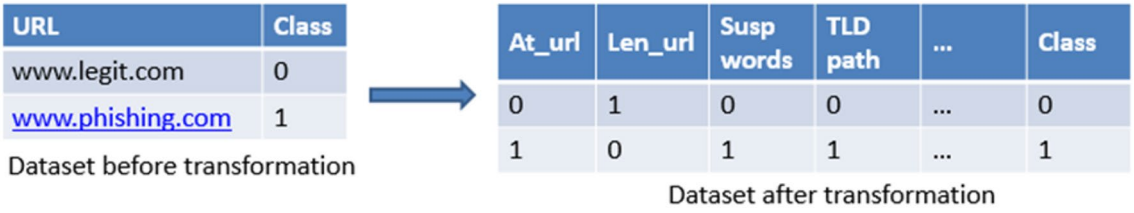


Fig. 5 Transformation process of the datasets into feature sets

Table 2 Datasets sizes and their sources

No	Dataset source	Dataset Size		
		Phishing	Legitimate	Total
D1	Kaggle[a]	1,14,203	3,92,801	5,07,004
D2	Kaggle[a]	55,914	39,996	95,910
D3	CatchPhish[b]	40,668	85,409	126, 077
D4	CatchPhish[b]	40,668	42,220	82,888
D5	CatchPhish[b]	40,668	43, 189	83,857
D6	Ebbu2017[c]	37,175	36,400	73,575

[a]https://www.kaggle.com/

[b]https://tinyurl.com/catchphish

[c]https://github.com/ebubekirbbr/pdd/tree/master/input

the literature used in Rao et al. (2019), and one dataset is collected from the literature Sahingoz et al. (2019). Distribution of phishing and legitimate URLs from these sources are shown in Table 2. All these datasets are open source and free to download.

3.2 Dataset preprocessing

Data preprocessing is the initial stage in any machine learning process, in which the dataset is transformed and encoded. In simple words, the datasets are converted into feature vectors, so then ML models can interpret the datasets easily. Moreover, after extraction of datasets, the preprocessing steps are applied on all the datasets used in this paper. The datasets have some noise and redundant URLs which needs to be removed from the datasets.

3.2.1 Data cleaning

In this step, noisy and redundant URLs have been removed from the datasets. The redundant URLs have been found in Kaggle D1 and D2 datasets, which is removed for further processing. There is some noise in the datasets, such as some URLs are in quoted form and some URLs are separated by commas. Thus, before extracting features from the URLs, these abnormalities needs to be removed from the datasets.

3.2.2 Feature engineering

Feature engineering is an essential step in any phishing detection method, as the performance of the method critically depends on it. Although features extracted in the past are extensive, with a limitation to take a considerable amount of time to extract such features. Thus, to overcome such issues, we have extracted features from URL string itself, which makes the process faster and efficient. For feature extraction, a python script is written, which takes dataset as an input and output the dataset in feature set. Initially, we have extracted 100 features from different parts of the URL and then selected top 30 features using domain knowledge and ReliefF technique in WEKA tool for training and testing the machine learning models to classify phishing and legitimate URLs. The extracted features are classified into three categories:

(a) Full URL-based features
(b) Host-based features
(c) Path area of the URL-based features

Table 3 provides a detail list of extracted features along with feature name and its description that is based on full URL, host and path area of the URL.

(a) *Full URL-based features* These features are extracted directly from the whole URL string. The novel features in this category are: F2–F4, F7–F8, F12 and the features F1, F5–F6, F9–F11 are adopted from the literature. Further, these features are divided into three categories:

- *Special characters-based features* There are several tokens and characters that occur more frequently in phishing URLs but missing in legitimate URLs. These features check the presence of characters such as (@, ~ ,!,^,*',(,),[,],{,},<,|,+,$,=,&,:,#,%) in full URL. The aforementioned features can be validated in entire URL, in hostname and in path section of the URL. We have also proposed new feature named as "ampersand greater than equal". This feature counts

Table 3 List of proposed 30 features

No	Feature name	Feature description
F1	At_url	Presence of "@" in the URL
F2	Amp_greater_equal	If number of "&" is greater than number of "=" in URL
F3	Delims_url	Presence of delimiters ('~', '`', '!', '^', '*', '(', ')', '[', ']', '{', '}', "", """, ';', ',', '>', '<', '|') in the URL
F4	Other_delims_url	Number of ('+', '$', '=', '&', ':', '#', '%') in URL
F5	Len_url	Length of the URL
F6	Email_exist	Presence of email address in the URL
F7	Protocol_url	Check if "http" and "www" words appears in the URL
F8	Suspwords_url	Presence of suspicious words in URL
F9	Digits_url	Number of digits in the URL
F10	Entropy_url	The higher the entropy, the more phishy the URL is
F11	Tiny_url	Presence of tiny urls in the URL
F12	Ratio_url_path	Ratio of full URL and path section of the URL
H1	Dot_host	Number of dots in the host
H2	Len_subdomain	Length of subdomain
H3	Having_https	Presence of HTTPS protocol in URL
H4	Brand_host	Presence of popular brand names in host
H5	Host_large_tok	Return the largest token in the host of the URL
P1	Path_large_tok	Return the largest token in the path of the URL
P2	TLD_path	Presence of TLDs in the path section of the URL
P3	Len_file	Length of name of the file section
P4	Extension	Presence of valid file and page extension names
P5	Delims_params	Presence of delimiters ('~', '`', '!', '^', '*', '(', ')', '[', ']', '{', '}', "", """, ';', ',', '>', '<', '|') in the parameters section of the URL
P6	Len_params	Length of parameter section
P7	Dot_path	Number of dots in the path
P8	Hyphen_path	Number of hyphens (-) in the path
P9	Slash_path	Number of slash (/) in the path
P10	Len_path	Length of the path
P11	Brand_path	Presence of brand name in the path
P12	Digits_path	Number of digits in the path
P13	Entropy_path	The higher the entropy of the path, the more phishy the URL is

the presence of ampersand (&) and equal (=) sign in the full URL, which further returns that if (&) count is greater than (=) then the URL is labelled as phishing otherwise legitimate. Features included in this category are F1F4, F6.

- *Count/Presence-based features* Some of the important features lie in this category. These features are used to count specific characters or record the length of the tokens or strings. Features included in this category are F5, F7, F8, F9, F11–F12. These features classify the URLs as phishing, if the length or count of specific feature gets higher.

- *Suspicious words-based features* We have compiled a list of suspicious words that have been used most frequently in phishing URLs. These words can be seen in Fig. 6. This feature check the presence of these words in the full URL string and if the suspicious word is present in the URL then it classifies as phishing otherwise legitimate. The feature included in this category is F8.

(b) *Host-based features* These features are extracted directly from the hostname, domain name or subdomain name of the URL. The novel feature in this category is H4, and the features H1–H3, H5 are adopted from the literature. Further, these features are divided into one category:

- *Count/Presence-based features* These features are used to count specific characters or record the length of the tokens or strings in hostname, domain and subdomain of the URL. Features included in this category are H1-H3, H5. These features classify the URLs as phishing, if the length or count of the specific feature gets higher. H4 feature is about the brand name matching which is discuss further in this section.

Fig. 6 Suspicious words to detect phishing URLs

(c) *Path section of URL-based features* Unlike extracting features from full URL, these features are extracted directly from the path section of the URL. The novel features in this category are: P2, P5, P11 and the features P1, P3-P4, P6-P10, P12-P13 are adopted from the literature. Further, these features are divided into two categories:

- *Special characters-based features* These features are similar to the ones discuss previously under the category of full URL based features. The only difference is that, these features are extracted from the path section of the URL and not from the full URL. These features check the presence of special delimiters in the path section of the URL. If these special characters found in the path of the URL, then the URL is labelled as phish otherwise legitimate. Feature included in this category is P5.
- *Count/Presence-based features* These features are similar to the ones discuss previously under the category of full URL and host based features. The only difference is that, these features are used to count specific characters or record the length of the tokens from the path section of the URL and not from the full URL. Features included in this category are P1-P4, P6-P10, P12. These features classify the URLs as phishing, if the length or count of specific feature gets higher.

Brand name-based features We have collected top 500 popular brand names from Alexa. The features included in this category are H4 and P11. We have used a technique named as TF-IDF for matching a brand name in the URL. The term frequency–inverse document frequency, or TF-IDF, is a quantitative metric that is used to rate the relevance of a word in a URL based on how frequently it appears in that URL. The idea behind this metric is that if a term appears frequently in a URL, it must be significant, and we should assign it a high score. However, if a term appears in too many other URLs, it is likely not a unique identifier, and we should give it a lower score. We have used a python packages such as TfidfVectorizer and NearestNeighbors. The function takes a URL and brand names as an input, which returns the score, if the score is lower than 10 and higher than 90 then we consider URL as phish otherwise the URL is labelled as legitimate.

Entropy-based features Entropy is a measure of uncertainty and we have used Shannon's entropy as a feature to classify phishing URLs. The features included in this category are F10 and P13. We have calculated the entropy of full URL and path section of the URL. This feature check the randomness in the URL and if the function return higher number of entropy, then it classifies URL as phish otherwise legitimate. We have calculated Shannon's entropy using the following equation:

$$H(x) = -\sum_{i=0}^{n} p(x_i) log_b p(x_i) \quad (1)$$

3.2.3 Normalization

When the features in the dataset have diverse ranges, normalization is a strategy used during data preparation to adjust the values of numeric columns in a dataset to use a

common scale. In this paper, we have used MinMaxScaler normalization that scales the dataset into 0–1 range.

3.3 Train/Test ratio

Train/Test is a method to measure the efficiency of ML models. In this paper, we have used 70% data for training the models and 30% data for testing the models.

3.4 Machine learning classifiers

In order to build model, we have used different machine learning classifiers in our framework. To evaluate the performance of the proposed features, we applied eight different machine learning classifiers such as Naïve Bayes (NB), Logistic regression (LR), Support vector machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (J48), AdaBoost (AB), Multilayer Perceptron (MLP) and Random Forest (RF); to train our proposed model and to test the efficiency of the proposed features. The key purpose of comparing different classifiers is to choose the best ML classifier that performs well in terms of efficiency, accuracy and precision.

Moreover, machine learning classifiers are predominant into phishing detection because of its dynamic structure, better accuracy and ability to detect zero day phishing URLs. However, the only trade-off using ML classifiers are that they are time consuming even on smaller datasets. To improve the efficiency of ML classifiers, one such approach is URL based phishing detection. Although URL provide limited information about the website but it has rich set of features that can detect phishing URLs smartly, timely and accurately.

Apart from the numerous advantages of ML classifiers, one of the most important prerequisites for ML is the availability of big phishing datasets for automated training. Hence, in this paper, ML framework is proposed that is feature efficient for URL phishing detection. The framework has utilized URL string for feature extraction and selection that is independent of any 3rd party services. Most of the past work depends only on the predetermined and 3rd party based features. While we explore different type of features and only focuses on the lexical based features, which is fast, accurate and efficient as compare to other approaches. The framework is evaluated on six different datasets and examined the results of eight state-of-the-art ML classifiers for phishing URL detection. To implement and experiment these various ML classifiers, WEKA tool is used and Python script is written for feature extraction. We found that RF and J48 outperformed other classifiers based on the experimental findings.

Decision Tree (J48) It is a supervised learning method that's commonly used to solve classification problems. It works well for both categorical and continuous variables. In this algorithm, population is divided into two or more homogenous sets and create as many unique groups as feasible based on the most important features. The advantages of DT are with simple interpretation and quick training. There is a drawback that over-fits the data and must be pruned to have been fixed with. It is inefficient in terms of speed and regression performance.

Random Forest (RF) An ensemble of decision trees is referred to as Random Forest (RF). RF has a collection of decision trees (also known as "Forest"). Each tree offers a categorization to a new object based on characteristics and votes for that class. The categorization with the highest votes is chosen by the forest. The following is how each tree is planted and grown:

- If the training set contains N instances, a sample of N cases is chosen at random but with replacement. This sample will serve as the set of tree-training.
- If there are M input variables, a number m is smaller than M is provided, so that m variables are randomly chosen from the M at each node and the best split on this m is used to divide the node. During the growth of the forest, the value of m is kept constant.
- Each tree is cultivated to its full potential. Pruning is not an option.

RF is a decision tree-generating binary and multiclass classifier. RF has the benefit of not over-fitting the data. It's quick and works well with huge datasets. It can handle a high number of features, which is useful for classification and regression. The drawback is that RF is un-interpretable and may over-fit the dataset when there is noise. On large datasets, it can use a lot of memory and cost a sufficient pecuniary burden. While training a large dataset may confront sluggishness.

3.5 Experimental and implementation setup

The experiments are conducted using python and WEKA tool. The datasets transformation and features are extracted using python script and the selection of features, evaluation of classifiers are performed in WEKA tool. The experiments are conducted on six different datasets using eight different machine learning classifiers. The datasets are divided into 70% for training, to train the model and 30% for testing the model.

3.6 Performance evaluation measures

Once the model is trained, the performance of the model is evaluated on testing dataset using different evaluation measures. The

performance evaluation is must for phishing URL detection as it determines the performance from different perspectives such as accuracy, precision, true positive rate and much more. The past work lacks in many of the performance measures such as FPR and precision which shows the efficiency of the model. We have trained the model on 70% of the dataset and tested the model on 30% of the dataset. The performance evaluation measures resulted an outcome of the best model that obtained the best results. Following are the performance measures used for evaluation of the models being used in this paper.

(1) **True Positive Rate (TPR)**

TPR indicates the number of phishing URLs that is classified as phishing. The highest the rate of TP, the more accurate the classifier can detect phishing URLs.

$$TPR = \frac{\text{No. of Phishing URLs classified as Phish}}{\text{Total no. of Phishing URLs}} \tag{2}$$

$$TPR = \frac{TPR}{TPR + FNR} \tag{3}$$

(2) **False Positive Rate (FPR)**

FPR indicates the number of legitimate URLs that is classified as phishing. The lower the rate of FP, the more accurate the classifier can detect phishing URLs.

$$FPR = \frac{\text{No. of Legitimate URLs classified as Phish}}{\text{Total no. of Legitimate URLs}} \tag{4}$$

$$FPR = \frac{FPR}{FPR + TNR} \tag{5}$$

(3) **True Negative Rate (TNR)**

TNR indicates the number of legitimate URLs that is classified as legitimate. The higher the rate of TN, the more accurate the classifier can detect legitimate URLs.

$$TNR = \frac{\text{No. of Legitimate URLs classified as Legit}}{\text{Total no. of Legitimate URLs}} \tag{6}$$

$$TNR = \frac{TNR}{TNR + FPR} \tag{7}$$

(4) **False Negative Rate (FNR)**

FNR indicates the number of phishing URLs that is classified as legitimate. The lower the rate of FN, the more accurate the classifier can detect legitimate URLs.

$$FNR = \frac{\text{No. of Phishing URLs classified as Legit}}{\text{Total no. of Phishing URLs}} \tag{8}$$

$$FNR = \frac{FNR}{FNR + TPR} \tag{9}$$

(5) **Precision**

Precision is the number of phishing URLs classified as phish divided by total number of phishing URLs.

$$Precision = \frac{\text{No. of Phishing URLs classified as Phish}}{\text{Total no. of URLs classified as Phish}} \tag{10}$$

$$Precision = \frac{TPR}{TPR + FPR} \tag{11}$$

(6) **F1-Score**

F1-Score or sometimes called as F1-Measure is, 2 multiplied by precision into recall divided by precision plus recall.

$$F1 - Score = \frac{2TPR}{(2TPR + FPR + FNR)} \tag{12}$$

(7) **ROC**

A receiver operating characteristic curve (ROC curve) is a graph that shows how well a classification model performs across all categorization levels. Two parameters are plotted on this curve: TPR and FPR.

(8) **Accuracy**

Accuracy indicates the correctly classified number of phishing and legitimate URLs. The highest the rate of accuracy, the more accurate the classifier can detect phishing and legitimate URLs.

$$Accuracy = \frac{\text{No. of correctly classified Phish and Legit URLs}}{\text{Total No. of URLs}} \tag{13}$$

$$Accuracy = \frac{TPR + TNR}{TPR + TNR + FNR + FPR} \tag{14}$$

4 Experimental results & discussion

This section provide experimental study and implementation of the proposed framework for phishing URL detection using machine learning classifiers on several datasets. The purpose of conducting ML experimental study on benchmark datasets is to validate and implement the proposed model on various datasets that have been collected from literature

and show that the proposed framework has improved the efficiency and accuracy for phishing URL detection.

All experiments have been conducted on Core i7 5th generation machine with 4 GB RAM. Python script has been developed for data preprocessing and extraction of features while WEKA tool is used for experimentation. A number of ML classifiers has been used to detect phishing URLs. Moreover, experiments are conducted on six different datasets using eight ML classifiers. The proposed framework is evaluated in terms of effectiveness and ability to classify phishing URLs successfully. Table 4 provides the details of tools and settings used for machine learning models.

During the classifier assessment stage, each of the built classifier is evaluated to see how good the proposed model is, in detecting phishing URLs. During this phase, a series of experiments are performed on the datasets, with 70% of the datasets being used for training and 30% for testing the model. Each dataset comprises a diverse group of URLs that are all distinct from one to another. In the next section, six different datasets are used to test ML classifiers for phishing URL detection.

4.1 Experiment 1: evaluation of classifiers on Kaggle (D1) dataset

This experiment was conducted on the bigger dataset named as Kaggle (D1). We have collected this dataset from the Kaggle repository and extracted 100 features from this dataset. Further evaluating the feature set dataset, we have selected the top ranked 30 features using domain knowledge and ReliefF technique in WEKA tool. After the selection of the features, we have applied eight different classifiers such as NB, LR, SVM, KNN, AdaBoost, MP, J48 and RF on this dataset. The goal of this experiment is to determine the best classifier for phishing detection among the various ML classifiers. The dataset was partitioned into 70% for training and 30% for testing the model. Table 5 provide the experimental results, in which Random forest outperformed other classifiers by achieving a highest accuracy of 96.25% and lowest false positive rate of 2.3%. Since this dataset was the bigger dataset in size amongst other datasets, the features selected in this dataset which is 30, will further extracted from other datasets.

Table 4 Common settings used for ML models

Classifiers	Extract features using	Tool	Settings	Train/Test ratio	No. of features selected
NB LR SVM KNN AdaBoost MP J48 RF	Python	WEKA	Default	70:30	30

4.2 Experiment 2: evaluation of classifiers on Kaggle (D2) dataset

In this experiment, we have extracted proposed 30 features from the dataset and evaluated on eight machine learning classifiers. The reason to evaluate all the classifiers again on this dataset, is to validate our proposed features and the classifier for detection of phishing URLs. This dataset has more phishing URLs than legitimate URLs unlike other datasets. Table 6 provide the experimental results, in which it is clear that Random forest again outperformed other classifiers by achieving a best accuracy of 94.65% and lowest false negative rate of 5.8%.

4.3 Experiment 3: evaluation of classifiers on CatchPhish (D3) dataset

This experiment was conducted on the benchmark dataset named as CatchPhish (D1). In order to check the efficiency of our proposed framework and proposed features, we have collected this dataset and further three more datasets. The goal of this experiment is to determine the best classifier for

Table 5 Experimental results of Kaggle (D1) dataset

Classifiers	TPR (%)	FPR (%)	TNR (%)	FNR (%)	Precision (%)	F1-score (%)	ROC (%)	Accuracy (%)
NB	35.5	4.6	95.4	64.5	74.2	63.5	82.6	66.97
LR	82.6	12.6	87.4	17.4	85.1	85.1	92.9	85.1
SVM	75.7	11.1	88.9	24.3	82.9	82.5	82.3	82.6
KNN	88.5	8.3	91.7	11.5	90.2	90.2	92.0	90.16
AdaBoost	73.7	14.7	85.3	26.3	80.0	79.7	85.0	79.8
MP	86.8	9.1	90.9	13.2	88.9	88.9	95.4	88.93
J48	91.2	6.8	93.2	8.8	92.3	92.3	94.9	92.3
RF	94.8	2.3	97.7	5.2	97.6	96.2	98.3	96.25

phishing detection among the various ML classifiers. Table 7 provide the experimental results, in which, it is clear again that Random forest outperformed other classifiers by achieving a highest accuracy of 92.2% and lowest false negative rate of 13.9%. Although, SVM achieved better false positive rate and true negative rate as compared to RF but SVM is expensive, when it comes to training the model and has achieved less accuracy of 82.84%.

4.4 Experiment 4: evaluation of classifiers on CatchPhish (D4) dataset

Similar to the previous experiment, this experiment is performed on the benchmark dataset, taken from literature. The goal of these experiments is to validate our framework in diverse datasets. The experimental findings of this dataset is shown in Table 8. From the results, Random forest again achieved better accuracy compared to other classifiers but achieved less accuracy as compared to other datasets. The accuracy came down from 96.25 to 91.63% on this dataset. The reason behind this is that the dataset has more common URLs that is extracted from the different sources such as Alexa. Although, the accuracy achieved is less but the ROC has reached to 97.2% as compared to D3 dataset.

4.5 Experiment 5: evaluation of classifiers on CatchPhish (D5) dataset

Similar to previous experiments, this experiment is also performed on the benchmark dataset, taken from the same

Table 6 Experimental results of Kaggle (D2) dataset

Classifiers	TPR (%)	FPR (%)	TNR (%)	FNR (%)	Precision (%)	F1-score (%)	ROC (%)	Accuracy (%)
NB	57.7	3.3	96.7	42.3	96.0	72.1	92.0	74.08
LR	89.0	7.4	92.6	11.0	94.3	91.5	96.5	90.48
SVM	86.5	5.9	94.1	13.5	95.3	90.7	90.3	89.69
KNN	92.1	8.9	91.1	7.9	93.4	92.8	93.0	91.67
AdaBoost	86.0	13.2	86.8	14.0	90.0	88.0	92.4	86.36
MP	90.2	5.8	94.2	9.8	95.6	92.8	97.2	91.86
J48	92.9	7.0	93.0	7.1	94.8	93.8	95.4	92.93
RF	94.2	4.8	95.2	5.8	96.4	95.3	98.6	94.65

Table 7 Experimental results of CatchPhish (D3) dataset

Classifiers	TPR (%)	FPR (%)	TNR (%)	FNR (%)	Precision (%)	F1-score (%)	ROC (%)	Accuracy (%)
NB	43.5	6.2	93.8	56.5	76.9	55.6	84.5	77.66
LR	66.0	6.0	94.0	34.0	83.9	73.9	90.7	85.01
SVM	53.7	3.4	96.6	46.3	88.3	66.8	75.2	82.84
KNN	81.1	8.4	91.6	18.9	82.1	81.6	87.5	88.24
AdaBoost	47.2	3.6	96.4	52.8	86.2	61.0	87.2	80.61
MP	82.7	8.0	92.0	17.3	83.1	82.9	94.1	89.05
J48	84.7	6.0	94.0	15.3	87.1	85.9	92.0	91.05
RF	86.1	4.9	95.1	13.9	89.3	87.6	96.3	92.2

Table 8 Experimental results of CatchPhish (D4) dataset

Classifiers	TPR (%)	FPR (%)	TNR (%)	FNR (%)	Precision (%)	F1-score (%)	ROC (%)	Accuracy (%)
NB	50	8.5	91.5	50.0	85.3	63.0	87.5	70.91
LR	82.6	13.3	86.7	17.4	86.0	84.2	92.4	84.65
SVM	78.1	11.4	88.6	21.9	87.1	82.3	83.3	83.37
KNN	86.7	14.2	85.8	13.3	85.8	86.3	86.9	86.28
AdaBoost	80.1	20.9	79.1	19.9	79.1	79.6	87.6	79.62
MP	87	12.4	87.6	13.0	87.4	87.2	94.5	87.32
J48	90.7	10.9	89.1	9.3	89.1	89.9	92.3	89.85
RF	91.5	8.2	91.8	8.5	91.7	91.6	97.2	91.63

Table 9 Experimental results of CatchPhish (D5) dataset

Classifiers	TPR (%)	FPR (%)	TNR (%)	FNR (%)	Precision (%)	F1-score (%)	ROC (%)	Accuracy (%)
NB	72.7	3.4	96.6	27.3	95.4	72.7	95.0	84.95
LR	90.0	5.1	94.9	9.1	94.4	92.6	97.0	92.93
SVM	91.5	5.7	94.3	8.5	93.9	92.7	92.9	92.96
KNN	91.8	5.5	94.5	8.2	94.1	92.9	93.9	93.19
AdaBoost	88.7	5.5	94.5	11.3	93.9	91.2	96.5	91.68
MP	94.4	5.9	94.1	5.6	93.9	94.1	97.8	94.27
J48	94.1	4.5	95.5	5.9	95.2	94.6	97.1	94.80
RF	94.6	5.0	95.0	5.4	94.7	94.7	97.9	94.80

literature as of D4 and D5. The experimental findings of this dataset is shown in Table 9. From the results, Random forest again achieved better accuracy of 94.80% compared to other classifiers and previous two datasets. The improvement in accuracy on this dataset is lead to validate our proposed features and framework.

4.6 Experiment 6: evaluation of classifiers on Ebbu2017 (D6) dataset

This experiment was conducted on the benchmark dataset named as Ebbu2017 (D6). In order to check the efficiency of our proposed framework and proposed features, we have collected this dataset from Sahingoz et al. (2019). The goal of this experiment is to determine the best classifier for phishing detection among the various ML classifiers. Table 10 provide the experimental results, in which, it is clear again that Random forest outperformed other classifiers by achieving a highest accuracy of 96.85% and lowest false negative rate of 2.6%. Although, NB achieved better false positive rate of 2.8% but it has achieved lesser accuracy of 64.86%. ROC of this dataset is 99.4%, which is far better than other datasets.

4.7 Experiment 7: comparison of proposed framework with past approaches

In this section, we have comparatively analyze the results of our best classifier with the past techniques such as Rao et al. (2019), Korkmaz et al. (2020), Chatterjee and Namin (2019), Al-Alyan and Al-Ahmadi (2020) that have utilized the same datasets for the classification of phishing URLs. Under the result set, comparative analysis with the past techniques has been performed. These approaches have detect phishing links based on URLs and not using any content or third party based techniques. Table 11 provide the comparison results based on different evaluation measures. Our framework outperformed existing techniques on CatchPhish (D3) with an accuracy of 92.2% and 4.9% of false positive rate. This indicates that our proposed features are much accurate and efficient on this dataset. Similarly, our framework has achieved 91.63% accuracy on CatchPhish (D4) dataset by outperforming the past techniques. Our framework has again achieved less FPR of 8.2% on this dataset whereas, the past techniques has achieved more than 10% of the FPR.

Moreover, there is a slight difference between our framework and the past approaches on CatchPhish (D5), in which we have achieved the accuracy of 94.80%, while past techniques have achieved 94.32% and 94.59% accuracy. The results of this dataset achieved is nearly same as the past techniques, with a slight difference. At last, our framework

Table 10 Experimental results of Ebbu2017 (D6) dataset

Classifiers	TPR (%)	FPR (%)	TNR (%)	FNR (%)	Precision (%)	F1-score (%)	ROC (%)	Accuracy (%)
NB	33.4	2.8	97.2	66.6	92.4	49.1	91.4	64.86
LR	91.9	10.5	89.5	8.1	90.0	90.9	97.0	90.7
SVM	92.1	13.2	86.8	7.9	87.7	89.9	89.4	89.46
KNN	95.5	6.2	93.8	4.5	94.0	94.7	94.9	94.63
AdaBoost	86.9	16.2	83.8	13.1	84.7	85.8	93.2	85.39
MP	93.6	8.2	91.8	6.4	92.2	92.9	97.8	92.72
J48	95.4	5.4	94.6	4.6	94.8	95.1	96.5	95.04
RF	97.4	3.7	96.3	2.6	96.5	96.9	99.4	96.85

Table 11 Comparison of proposed framework with the existing techniques

Approaches		TPR (%)	FPR (%)	TNR (%)	FNR (%)	Precision (%)	F1-score (%)	ROC (%)	Accuracy (%)
Datasets	Techniques								
CatchPhish (D3)	Rao et al. (2019)	91.67	12.90	87.10	8.33	94.22	92.93	X	90.28
	Korkmaz et al. (2020)	93.02	12.47	87.53	6.98	88.18	90.53	x	91.26
	Proposed method	86.1	4.9	95.1	13.9	89.3	87.6	96.3	92.2
CatchPhish (D4)	Rao et al. (2019)	88.60	10.69	89.31	11.4	89.86	89.23	x	88.95
	Korkmaz et al. (2020)	91.11	10.1	89.9	8.89	90.02	90.56	x	90.50
	Proposed method	91.5	8.2	91.8	8.5	91.7	91.6	97.2	91.63
CatchPhish (D5)	Rao et al. (2019)	94.41	5.79	94.2	5.59	94.56	94.49	x	94.32
	Korkmaz et al. (2020)	94.59	5.51	94.49	5.31	94.5	94.59	x	94.59
	Proposed method	94.6	5.0	95.0	5.4	94.7	94.7	97.9	94.80
Ebbu2017 (D6)	Rao et al. (2019)	96.38	4.81	95.19	3.62	95.02	95.69	x	95.77
	Chatterjee and Namin (2019)	88.0	x	x	x	86.7	87.3	x	90.1
	Al-Alyan and Al-Ahmadi (2020)	x	x	x	x	96.5	x	x	93.64
	Proposed method	97.4	3.7	96.3	2.6	96.5	96.9	99.4	96.85

has achieved better accuracy of 96.58% and ROC of 99.4% on the dataset of Ebbu2017 (D6). From the results, we have outperformed all the past techniques with a significant accuracy, false positive rate and ROC.

5 Discussion and limitations

This paper was motivated by the fact that the phishing URLs are increasing rapidly and there isn't any single approach that can handle this threat easily and efficiently. Phishing is a persistent and major threat on the internet that is growing steadily and dangerously. It is a type of cyber-attack, in which phisher mimics a legitimate website page to harvest victims sensitive information such as usernames, emails, passwords, bank accounts and credit card details. To overcome the said issue, a number of phishing detection models has been proposed in the past such as AI based, 3rd party, heuristic and content based. However, the previously proposed techniques suffered from a number of limitations, which have been explored in this paper. Thus, to overcome said limitations, this paper proposed effective machine learning based framework for phishing URL detection. The proposed framework extracts only lexical based features, which means that the features are directly extracted from the URL string rather than it depends on any 3rd party and web-page source code services. The proposed framework has extensive amount of features by examining the legitimate and phishing URLs in depth and proposed 30 important features that has enhanced the efficiency, accuracy and performance for the detection of phishing URLs. The proposed framework is validated on six different datasets and results have been critically evaluated to comprehend the accuracy and effectiveness of the proposed framework.

In terms of model building, firstly, feature engineering is essential step in any phishing detection method, as the performance of the method critically depends on it. Although features extracted in past work are extensive, a limitation is that it takes a considerable amount of time to extract such features. Similarly, using features such as third party based produces high false positive rates. Secondly, several approaches selected important features using statistical methods while some approaches proposed their own features. Although both methods have been implemented successfully in various approaches, however, these methods produce incorrect results without amplification of domain knowledge. Thirdly, most of the literature has used pre-classified and smaller datasets for classification and evaluation of proposed models, which fails to produce the exact efficiency and precision on large and real world datasets. Fourthly, previously proposed approaches lack in advanced evaluation measures that can determine the performance of particular model for phishing detection.

In order to address the aforementioned limitations, a smart and accurate framework has been proposed in this paper, which overcome these limitations in an effective way. The proposed framework has been validated using ML classifiers. Experimental results have been conducted separately for ML models on six different datasets, taken from literature and Kaggle repository and efficiency of the proposed framework has been observed by accurate phishing URL detection.

Moreover, in this paper, we comparatively analyze the machine learning approaches such as Naïve Bayes (NB),

Logistic regression (LR), Support vector machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (J48), AdaBoost (AB), Multilayer Perceptron (MP) and Random Forest (RF) against their performance for phishing URL detection. The experiments carried out on several benchmark datasets by splitting them into 70% for training and 30% for testing the model. The comparative analysis with the existing approaches shows that the proposed model has outperformed other past techniques by obtaining better efficiency and accuracy. Furthermore, the comparative analysis reveals that RF is the only model that are superior to other models for detection of phishing URLs. Moreover, the performance of the RF has been comparatively analyzed with the past techniques in which RF outperformed other approaches by achieving better accuracy and low false positive rates.

We have also examine the training time of the classifiers that has been used in experiments on all the datasets. The training time is differentiating from smaller to larger datasets. Figure 7 provide a training time of all the eight classifiers on six different datasets. It is clear that the least training time is achieved by KNN and NB but there performance measures are not good as the time to build the model. RF is the best model in terms of accuracy, efficiency and performance, while its training time on all the datasets is relatively high. On the smaller dataset, it trains faster but on the larger dataset, which is D1, it takes 384 s to build the model. As compared to MP and SVM, RF training time is better but has higher build time than other classifiers. Moreover, we have mentioned earlier that each performance measure for evaluating the model is important, hence, RF has achieved better accuracy, FPR and ROC on all the datasets by outperforming all the classifiers.

Limitations Despite the fact that our suggested framework has a high level of accuracy, it has certain flaws. The framework first limitation is that, because the features are taken from the URL, it may misclassify some phishing URLs hosted on free hosting providers. Phishing URLs exist that do not match the patterns of known phishing sites. This might lead to such URLs being misclassified as valid.

The second flaw with this method is that it ignores any visual mimic behaviour that may exist in the source code or graphics. As a result of this, some URLs that are distinct from others yet closely resemble the target websites may not be recognized based on the URL string.

6 Conclusion

Phishing is a persistent and major threat on the internet that is growing steadily and dangerously. It is a type of cyber-attack in which phisher mimics a legitimate website page to harvest victims sensitive information such as usernames, emails, passwords, bank accounts and credit card details. To overcome this issue, this paper proposed effective Machine Learning based framework for phishing URL detection. Our framework is useful to empower the sector of

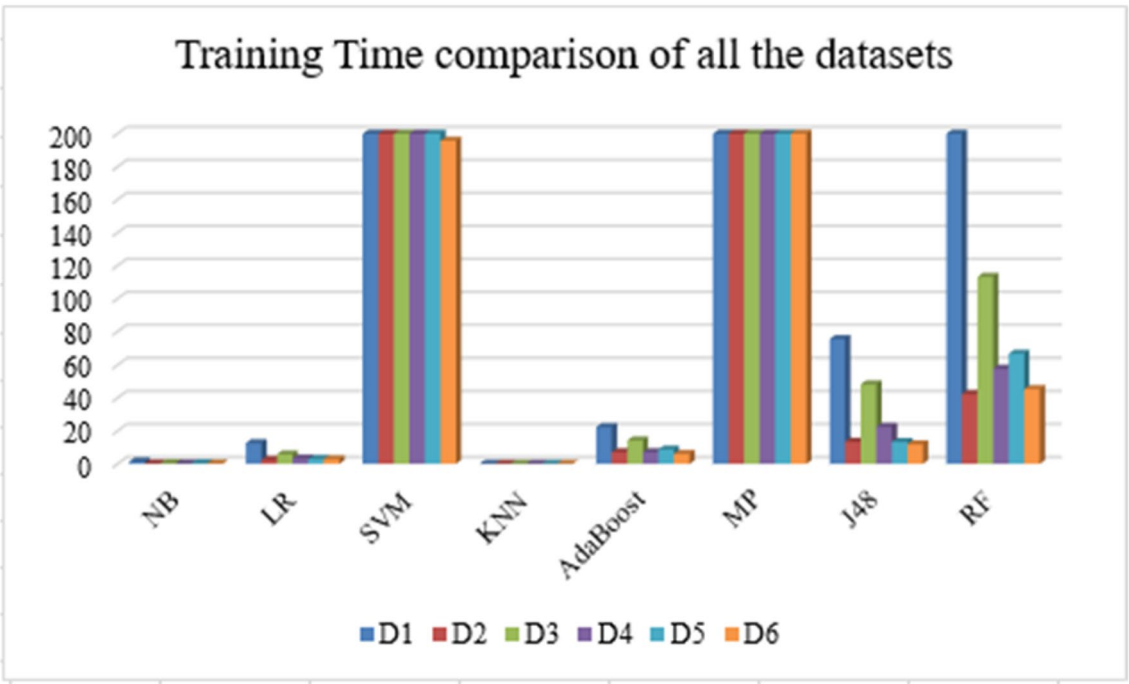


Fig. 7 Training time comparison

phishing detection and can gain further knowledge to stop this threat by harming normal day user. The proposed framework extracts only lexical based features which means that features are directly extracted from the URL string rather than it depends on any 3rd party or web-page source code and similarity. The proposed technique is based on URL and uses full URL, protocol scheme, hostname, path area of the URL, entropy feature, suspicious words and brand name matching using TF-IDF technique for the classification of phishing URLs. The experiments are carried out on six different datasets using eight different machine learning classifiers, in which Random Forest achieved a significant higher accuracy than other classifiers on all the datasets. The proposed framework with only 30 features achieved a higher accuracy of 96.25% and 94.65% on the Kaggle datasets. The comparative results show that the proposed model achieved an accuracy of 92.2%, 91.63%, 94.80, 96.85% on benchmark datasets, which is higher than the existing approaches.

In future, we intend to test the proposed framework on real world datasets, to comprehend the efficiency and accuracy of our framework. Moreover, we intend to explore more relevant features and select less features to detect phishing URLs more efficiently and accurately.

The classification phase of the proposed framework comprises of certain steps performed annually such as feature extraction and selection, pre-processing and conversion of the datasets. The feature extraction and preprocessing steps involved python script whereas the classification of phishing URLs are performed using tools such as WEKA. These steps, involve the manual selection of each phase but in future, we intend to build a smart tool that can take a URL as an input and output the prediction result whether the given URL in fake or legitimate.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12652-022-04426-3.

References

Aburub F, Hadi W (2021) A new association classification based method for detecting phishing websites. J Theoret Appl Inf Technol 99(1):147–158

Abuzuraiq A, Alkasassbeh M, Almseidin M (2020) Intelligent methods for accurately detecting phishing websites. In: 1th International Conference on information and communication systems (ICICS), pp 085–090, April 2020.

Al-Alyan A, Al-Ahmadi S (2020) Robust URL phishing detection based on deep learning. KSII Trans Internet Inf Syst 14(7):2752–2768

Alexa (2022) Most popular legitimate URLs. https://www.alexa.com/. Accessed 5 Aug 2021

Alsharnouby M, Alaca F, Chiasson S (2015) Why phishing still works: user strategies for combating phishing attacks. Int J Hum Comput Stud 82:69–82

APWG (2013–2020) Phishing activity trends reports, 1st, 2nd, 3rd, and 4th quarters of each years. https://apwg.org/trendsreports/, published 2013–2020

Bahnsen AC, Bohorquez EC, Villegas S, Vargas J, González FA (2017) Classifying phishing URLs using recurrent neural networks. In: IEEE Proceedings of the APWG Symposium on electronic crime research (eCrime), pp 1–8, 2017

Banik B, Sarma A (2018) Phishing URL detection system based on URL features using SVM. Int J Electron Appl Res (IJEAR) 5(2):40–55

Chatterjee M, Namin AS (2019) Detecting phishing websites through deep reinforcement learning. In: IEEE Annual Computer Software and Applications Conference, pp 227–232, 2019

Chavan S, Inamdar A, Dorle A, Kulkarni S, W, X-W (2019) Phishing detection: malicious and benign websites classification using machine learning techniques. In: Springer Proceeding of International Conference on computational science and applications (ICCSA), pp 437–446, August 2019

Chiew KL, Yong KSC, Tan CL (2018) A survey of phishing attacks: their types, vectors and technical approaches. Elsevier Expert Syst Appl 106:1–20

Chiew KL, Tan CL, Wong K, Yong KS, Tiong WK (2019) A new hybrid ensemble feature selection framework formachine learning-based phishing detection system. Inf Sci 484:153–166

Dou Z, Khalil I, Khreishah A, Al-Fuqaha A, Guizani M (2017) Systematization of knowledge (SoK): a systematic review of software-based web phishing detection. IEEE Commun Surveys & Tutor 19(4):2797–2819

El Aassal A, Baki S, Das A, Verma RM (2020) An indepth benchmarking and evaluation of phishing detection research for security needs. IEEE Access 8:22170–22192

Feng F, Zhou Q, Shen Z et al (2018) The application of a novel neural network in the detection of phishing websites. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-018-0786-3

Gupta BB, Yadav K, Razzak I, Psannis K, Castiglione A, Chang X (2021) A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. Comput Commun 175:47–57

Hutchinson S, Zhang Z, Liu Q (2018) Detecting phishing websites with random forest. Springer ICST Inst Comput Sci Soc Inf Telecommun Eng MILICOM 251:470–479

Jagadeesan S, Chaturvedi A, Kumar S (2018) Url phishing analysis using random forest. Int J Pure Appl Math 118(20):4159–4163

Jain AK, Gupta BB (2018a) PHISH-SAFE: URL features-based phishing detection system using machine learning. In: Springer cyber security, advances in intelligent systems and computing, pp 467–474

Jain AK, Gupta BB (2018b) A machine learning based approach for phishing detection using hyperlinks information. Springer J Ambient Intell Humaniz Comput, pp 2015–2028

Jalil S, Usman M (2020) A review of phishing URL detection using machine learning classifiers. Springer Adv Intell Syst Comput 1251:646–665

Jeeva C, Rajsingh EB (2016) Intelligent phishing url detection using association rule mining. SpringerOpen Human-Centric Comput Inf Sci 6:10

Joshi A, Pattanshetti TR (2019) Phishing attack detection using feature selection techniques. In: Proceedings of International Conference on communication and information processing (ICCIP), May 2019, pp 949–952

Korkmaz M, Sahingoz OK, Diri B (2020) Detection of phishing websites by using machine learning-based URL analysis. In: IEEE 11th International Conference on computing, communication and networking technologies (ICCCNT), pp 1–7

Kulkarni A, Brown LL (2019) Phishing websites detection using machine learning. Int J Adv Comput Sci Appl (IJACSA) 10/7:8–13

Li JH, Wang SD (2017) Phishbox: an approach for phishing validation and detection. In: 2017 IEEE 15th Int. Conf. on Dependable, Autonomic and Secure Computing, 15th Int. Conf. on Pervasive Intelligence and Computing, 3rd Int. Conf. on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Orlando, FL, USA., 6 November 2017, pp 557–564

Li Y, Yang Z, Chen X et al (2019) A stacking model using URL and HTML features for phishing webpage detection. Elsevier Future Gener Comput Syst 94:27–39

Opara C, Wei B, Chen Y (2020) HTMLPhish: enabling phishing web page detection by applying deep learning techniques on HTML analysis. In: IEEE International Joint Conference on neural networks (IJCNN), pp 1–8, 2020

Pandey A, Gill N, Sai Prasad Nadendla K, Sumaiya Thaseen I (2019) Identification of phishing attack in websites using random forest-SVM hybrid model. In: Springer intelligent systems design and applications (ISDA), pp 120–128

PhishTank (2022) Verified phishing URLs. https://www.phishtank.com/. Accessed 5 Aug 2021

Rao RS, Vaishnavi T, Pais AR (2019) CatchPhish: detection of phishing websites by inspecting URLs. Springer J Ambient Intell Humaniz Comput 11:813–825

Sadique F, Kaul R, Badsha S, Sengupta S (2020) An automated framework for real-time phishing URL detection. In: IEEE 10th annual computing and communication workshop and conference (CCWC), pp 0335–0341

Sahingoz OK, Buber E, Demir O, Diri B (2019) Machine learning based phishing detection from URLs. ScienceDirect J Expert Syst Appl 117:345–357

Shahrivari V, Darabi MM, Izadi M (2020) Phishing detection using machine learning techniques. arXiv 2009.11116

Srinivasa Rao RS, Pais AR (2018) Detection of phishing websites using an efficient feature-based machine learning framework. Springer Neural Comput Appl 31:3851–3873

Tan CL, Chiew KL, Wong K, Sze SN (2016) PhishWHO: phishing webpage detection via identity keywords extraction and target domain name finder. Elsevier Decis Support Syst 88:18–27

UCI (2022) UC Irvine Machine Learning Repository. https://archive.ics.uci.edu/ml/index.php/. Accessed 5 Aug 2021

Webroot (2020) Webroot threat report. https://mypage.webroot.com/rs/557-FSI-195/images/2020%20Webroot%20Threat%20Report_US_FINAL.pdf. Accessed 5 Aug 2021

Yang P, Zhao G, Zeng P (2019) Phishing website detection based on multidimensional features driven by deep learning. IEEE Access J Mag 7:15196–15209

Zhu E, Chen Y, Ye C, Li X, Liu F (2019) OFS-NN: an effective phishing websites detection model based on optimal feature selection and neural network. IEEE Access J Mag 7:73271–73284