IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

IEEE Access Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022. Doi Number

Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions

Malak Aljabri[1,2], Hanan S. Altamimi[2], Shahd A. Albelali[2], Maimunah AL-Harbi[2], Haya T. Alhuraib[2], Najd K. Alotaibi[2], Amal A. Alahmadi[3], Fahd Alhaidari[3], Rami Mustafa A. Mohammad[4], and Khaled Salah[5]

[1] Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia
[2] SAUDI ARAMCO Cybersecurity Chair, Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
[3] SAUDI ARAMCO Cybersecurity Chair, Department of Networks and Communications, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
[4] SAUDI ARAMCO Cybersecurity Chair, Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia
[5] Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, 127788, UAE

Corresponding author: Malak Aljabri (e-mail: msaljabri@iau.edu.sa).

This research was funded by SAUDI ARAMCO Cybersecurity Chair at Imam Abdulrahman Bin Faisal University (IAU).

ABSTRACT In recent years, the digital world has advanced significantly, particularly on the Internet, which is critical given that many of our activities are now conducted online. As a result of attackers' inventive techniques, the risk of a cyberattack is rising rapidly. One of the most critical attacks is the malicious URL intended to extract unsolicited information by mainly tricking inexperienced end users, resulting in compromising the user's system and causing losses of billions of dollars each year. As a result, securing websites is becoming more critical. In this paper, we provide an extensive literature review highlighting the main techniques used to detect malicious URLs that are based on machine learning models, taking into consideration the limitations in the literature, detection technologies, feature types, and the datasets used. Moreover, due to the lack of studies related to malicious Arabic website detection, we highlight the directions of studies in this context. Finally, as a result of the analysis that we conducted on the selected studies, we present challenges that might degrade the quality of malicious URL detectors, along with possible solutions.

INDEX TERMS Phishing; URL; Machine Learning; Cybersecurity; Random Forest; Malicious

I. INTRODUCTION

As the Internet develops and grows, many of our activities are now conducted online, including e-commerce, business, social networking, and banking, raising the likelihood of online crime. So, securing the world wide web is becoming increasingly important. According to Internet World Stats [1], around 237,418,349 users used the Arabic language on the Internet in 2020. Attempts to bait users to click through to malicious uniform resource locators (URLs) lead to the system being hacked or access being gained to sensitive data. Consequently, it is becoming increasingly necessary to secure this side. Protocols and regulations secure the connection between the client and server, yet it is still vulnerable to those with malicious intent to attack it. The term "Malicious" is a general term for attack types that include phishing, spam and malware, and more.

Malicious URLs are used to extract unsolicited information and trick inexperienced end users into falling for a scam, which causes losses of billions of dollars each year.

In order to identify the threat from malicious sites, the online security community has created blacklisting services to help detect harmful websites. The blacklist is a database that contains a list of all URLs already known to be malicious. URL blacklisting has been shown to be effective in some cases [2]. However, the attacker can make use of them by easily fooling the system with changes to one or more components of the URL string. Inevitably, many

malicious sites are not blacklisted because they are either too new or were never or erroneously assessed.

Another approach to identifying malicious sites is the heuristic method, which is an improved version of the blacklist method but based on signatures that are used to find the correlation between the new URL and the signature of an existing malicious URL. These approaches are adequate for identifying malicious and benign URLs. However, these previous methods have limitations, such as (a) the blacklist method failing to protect against zero-hour phishing attacks, as it classifies only 47–83% of new phishing URLs in a 12-hour period [3], and as a result, it cannot categorize new URLs [4], and (b) these methods can be bypassed using an obfuscation method, such as generating a huge number of URLs with an algorithm that can bypass the blacklist and heuristic methods due to failures with handling extensive lists, in which case the blacklist method cannot be used with rapid change technology. Despite these limitations, the blacklist method is used by many anti-phishing companies due to its simplicity.

The third approach to detecting these malicious sites is the use of artificial intelligence (AI) approaches, including machine learning (ML) and deep learning (DL). These approaches have been widely applied in different fields, including cybersecurity, healthcare, medical imaging analysis, e-commerce, and social media. Particularly in the cybersecurity field, it can take advantage of how ML models can be designed to learn from their previous experience and thus have better self-learning without the need for human interaction. This results in significant property in large organizations, companies, banks, and others. Moreover, ML and DL techniques have proved their ability in many disciplines, and they are frequently used to detect malicious sites [5].

The use of ML for detecting malicious URLs has proved to be effective through detecting newly formed URLs and the automatic update of the model. Recent studies have explored DL models that use an approach to automatically detect newly formed URLs and extract the features. In this way, researchers can extract many features from URLs that help ML algorithms categorize the URL as malicious or benign. The most common features extracted from the URLs are lexical, content-based, and net-work-based, as described next.

In this research, we reviewed 91 studies published from 2012 to 2021 that used ML or DL in the classification of malicious URLs. The contents of the websites were classified as either Arabic or English language. We provide a taxonomy of the reviewed studies on the detection of malicious URLs in terms of several aspects, including the language used, related URL features, ML detection techniques, and the datasets used. The primary contributions of this paper can be summarized as follows:

- Produces several taxonomies of malicious URL detection studies.
- Conduct many comparisons and discusses several techniques and properties related to malicious URL attacks and techniques for detecting them.
- Highlights several findings about the features of a URL that are used for detection, including the type of content, algorithms used for detection, and datasets used.
- Discusses several challenges that might impact the quality of ML detection techniques, including the size of the dataset, outliers, features selection, and the sustainability of the detectors.

The rest of this paper is organized as follows. Section 2 presents the background of URL feature types and the attack techniques. Section 3 provides the taxonomy of the works investigated in this study. Section 4 discusses and summarizes the ML studies about malicious URL attack detection. A discussion of the datasets used for evaluating detection techniques is presented in Section 5. Finally, Section 6 concludes the paper and presents future related work.

II. BACKGROUND

This section explains the common URL feature types and the possible types of attacks that can be used by attackers through URLs. The URL features discussed in this section include lexical, content, and network features. This section also discusses the common techniques of spam, phishing, malware, and defacement URL attacks.

A. URL FEATURES

The success of any ML model depends on the quality of training data and the quality of features fed into the model. Certain features must be available to analysts in order to create proactive models to identify malicious URLs. Simple URL strings can be used to extract these features, which can be lexical, content, or network [6],[7].

First, lexical features include the elements of the URL string. They are determined by how the URL looks or seems different in users' eyes and the URL's textual properties. These include statistical properties such as the length of the URL, length of the domain, number of special characters, and number of digits in the URL. Second, content features refer to the actual content on the page. These features are obtained upon opening or downloading the website, and it includes the hypertext markup language (HTML) tag count, Iframe count, hyperlink count, number of scripts, and count of suspicious JavaScript and other functions. Third, network features are a union of the domain name system (DNS), network, and host features. It also includes the resolved IP count, latency, redirection count, domain lookup time, number of DNS, connection speed, and the number of open ports.

The purpose of including these types of features is to enhance model performance to accurately detect malicious URLs. In general, it has been found that legitimate websites have more content than malicious websites. Moreover, network features can be useful in detecting malicious websites that tend to be hosted by less reputable service providers. Therefore, the DNS information can be used to detect malicious websites. Keywords extracted from the domain name can be compared to a list of commonly used keywords associated with malicious behavior. All of the mentioned features help in determining whether a web page is malicious [8].

B. URL ATTACK TECHNIQUES

Attack techniques are the methods or mechanisms used by attackers to illegally gain access to user data or cause damage to the attacked system. Attackers can use malicious URLs to perform those attacks. Malicious URLs can be classified as spam, phishing, malware, or defacement URLs.

The majority of cyberattacks happen when users click on malicious URLs. When URLs are exploited for purposes other than accessing legitimate resources on the Internet, they pose a threat to data integrity, confidentiality, and availability. The different kinds of malicious URLs are discussed below [9].

1) SPAM URL ATTACKS

These attacks occur when spammers create web pages in an attempt to fool the browser engine into perceiving they are legitimate when they are not. By illegally improving their rank, spammers want to deceive and attract more users to their spam websites [10]. Spammers send spam emails that contain spam URLs to harm and infect the systems of their victims using spyware and adware [11].

2) PHISHING URL ATTACKS

Attackers use phishing URLs to attract users to open a fake website, where access to the user's computer is attempted in order to steal a user's private information, such as credit card numbers. Non-expert users can be easily fooled into clicking through to a phishing website by making barely noticeable misspellings in the URL, such as changing www.facebook.com to www.facebo0k.com, which makes user data more vulnerable [11].

3) MALWARE URL ATTACKS

These attacks direct users to a malicious website that typically installs malware on the user's device that can be exploited for file corruption, keystroke logging, and even identity theft. Malware is a type of malicious software that can steal someone's personal information and damage a computer. One example of malware is the drive-by download, defined as the unintentional download of malware caused by a user being tricked into visiting a malicious website [12]. More examples include ransomware, keyloggers, trojan horses, spyware, scareware, computer worms, and viruses [11].

4) DEFACEMENT URL ATTACKS

This type of attack redirects the user to a malicious website that has been altered by hackers in one or more aspects, such as its visual appearance or some of the site's contents. Hacktivists strive to take down a website for several reasons [13]. This form of action occurs when the attackers discover the vulnerabilities of the website and utilize those vulnerabilities to compromise the website and modify the content on the web page without the owner's authorization, which is technically known as penetrating a website [11].

The classification of malicious URL attacks by ML techniques can be binary, such as either malicious or benign. Conversely, multi-classification is not restricted to any number of classes except that it has more than two, such as benign, phishing, suspicious, malware, spam, and others.

III. TAXONOMY OF MALICIOUS URL DETECTION TECHNIQUES

The existing works investigated in this research encompassed studies conducted between 2012 and 2021. Figure 1 provides a complete view of the explored studies based on the detection of malicious URLs according to ML detection techniques, classification types, and used datasets.

We examined and summarized related work in the detection of malicious URLs on Arabic and English websites using ML algorithms. The type of classification for each study and the name of the classifications are exactly as written in the study. In general, most of the studies used binary classification of URLs. Overall, 81 studies used binary classification, and 10 studies used multi-classification. The datasets utilized to train and test the detection models in the examined studies came from a variety of sources, including open sources, those created by the study authors, those adapted from other authors, or a combination. The most common dataset sources were PhishTank[14] and Alexa, as well as datasets collected by the study authors. ML algorithms can be classified into three

categories: supervised learning, unsupervised learning, and semi-supervised learning, which refer to labelled, unlabelled, and partially labelled training data, respectively.

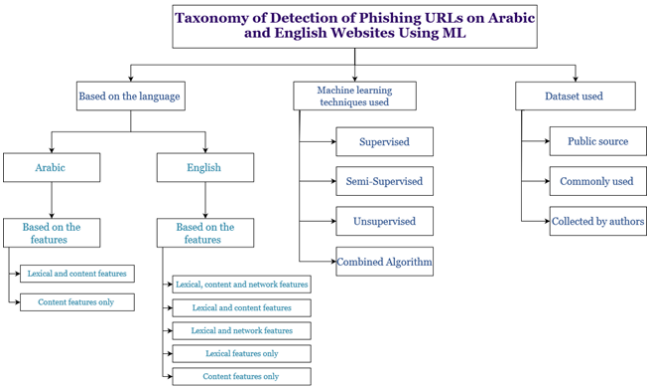

FIGURE 1. Taxonomy of malicious URL detection on Arabic and English websites using machine learning (ML).

IV. MALICIOUS URL DETECTION ON ARABIC AND ENGLISH STUDIES

This section reviews and summarizes the related work in terms of detecting Arabic and English malicious attack websites using ML algorithms. Many features can be extracted from the URL to help ML algorithms accurately detect malicious URLs. The features mentioned in the reviewed studies were sorted according to three main features in order to use unified terms in the present research. Those main features are lexical, content-based, and network-based, as discussed in Section 2. The summarized papers have been further linked based on the type of features they used and whether the website contents were in Arabic or English.

A. ENGLISH-BASED STUDIES

Many studies have been conducted investigating different features and algorithms for malicious attack detection on English content websites. This section presents these studies and categorizes them into five sections. The first section presents studies that focused on lexical, content, and network-based features. The second section presents studies that focused on lexical and content-based features. The third section presents studies that focused on lexical and network-based features; the fourth section presents studies that focused on lexical features only; and the fifth section presents studies that focused on content-based features only.

1) LEXICAL, CONTENT-BASED, AND NETWORK-BASED FEATURES STUDIES

The research conducted by Aldwairi et al. [15] was based on a lightweight self-learning scheme. The open-source datasets used were Alexa, which contains benign websites [16], and the PhishTank dataset, which contains malicious URLs [14]. The extracted features are lexical, network-based, and content-based, with a total of 31 features. The system achieved a precision of 87%.

Another study, which was conducted by Xuan et al. [17], proposed a malicious URL detection method using ML techniques. To classify URLs, they used 54 lexical, net-work-based, and content-based features. Their random forest (RF) algorithm resulted in high accuracy at 96.28%. However, Molah et al. [18] achieved better accuracy of 97.36% using RF in an intelligent system for detecting phishing websites using different ML techniques. Their dataset was adopted from the University of California Irvine Machine Learning Repository (UCI–ML) [19]. A total of 30 lexical, network, and content-based features were extracted to classify the URLs.

Yuan et al. [20], proposed a parallel neural joint model algorithm to analyse and detect malicious URLs by combining the technologies of a labelled capsule neural network (CapsNet) and independently recurrent neural network (IndRNN). The dataset used was collected from PhishTank [14], Malware Domain List [21], and Alexa [22]. They used lexical and other features, and their model included three parts: IndRNN, CapsNet, and attention. Their proposed method achieved an accuracy of 99.78%.

In addition, Yu [23] proposed a hybrid model that combined the advantages of a deep belief network (DBN) and support vector machine (SVM) for phishing website detection. The dataset was collected from PhishTank [14]. The features considered were lexical, content-based, and network-based. The model (DBN-SVM) achieved the highest accuracy of 99.96%.

Another study formulated by Zamir et al. [24] proposed a framework to detect phishing websites using a stacking model. The used dataset is Kaggle [25]. They extracted 32 content, lexical, and network features. Two stacking models were formed based on the highest scoring classifiers: Stacking 1 (RF + neural network (NN) + bagging classifier (BC)) and Stacking 2 (K-nearest neighbor (KNN) + RF + BC). The highest achieved accuracy was 97.4% with the Stacking 1 model (RF + NN + BC).

A different approach was used by Alkhudair et al. [26], who applied a malicious URL detection method using four ML algorithms. They obtained their dataset from the Kaggle [27] and Urcuqui et al. [28] datasets and used 20 lexical, content-based, and network-based features. RF had the best result, with 95% accuracy.

However, Deebanchakkarawarthi et al. [29] achieved better accuracy of 97% in their ML methodology aiming to avoid database dependency, increase efficiency, and detect malicious URLs.

In order to detect and categorize malicious URLs Selvaganapathy et al. [30].proposed a methodology based on a stacked restricted Boltzmann machine for feature selection with deep NN. The dataset was formed from MalwareDomainList; UCI–ML Repository: Spambase Dataset [31]; UCI–ML Repository: Phishing Dataset [19]; DMOZ [32]; and Alexa [16]. A total of 98 features were extracted. The highest accuracy was achieved by DBN (75%).

Similarly, Rao et al. [33] proposed a heuristic technique to detect phishing sites hosted on compromised servers. The dataset was collected from the PhishTank website [14] and Alexa [22]. They selected 6 lexical features, 1 network-based feature, and 10 content-based features. The highest accuracy of 98.05% was achieved by the twin SVM (TWSVM).

Additionally, Vinayakumar et al. [34] reviewed the effectiveness of various DL mechanisms for detecting malicious URLs. They built two datasets, the first formed from MalwareURL, MalwareDomains (now Risk Analytics) [35], PhishTank [14], OpenPhish [36], and MalwareDomainList [21]. The second dataset was formed from Alexa [22] and the DMOZ directory [32]. The extracted features were lexical, content-based, and network-based. The long short-term memory (LSTM) algorithm outperformed the others with 99.96% accuracy.

Patil et al. [37] proposed a methodology to detect malicious URLs and the type of attacks based on multi-class classification. The dataset was collected from the Alexa top sites and PhishTank [14], MalwareDomainList [21], and jwSpamSpy [38]. They extracted 65 lexical, 34 content-based, and 18 network-based attacks. The highest average accuracy of 98.44% in identifying the attack type was achieved by the confidence-weighted (CW) learning classifier. In the detection of malicious URLs, they achieved an accuracy of 99.86%. A limitation of their methodology is that it lacks the detection and analysis of obfuscated JavaScript on web pages.

Yang et al. [39] presented multidimensional feature phishing detection (MFPD) based on a DL detection method. They created a dataset by crawling PhishTank [14] and DMOZ [40], and the three types of URL features selected were lexical, content-based, and network-based. The MFPD algorithm achieved the best performance of 98.99% accuracy.

In addition, Mourtaji et al. [41] proposed a hybrid rule-based methodology to detect and control phishing websites. Their dataset was collected from PhishTank [14] and Alexa [16], and they extracted 37 lexical, network-based, and content-based features. The best accuracy was achieved by the convolutional NN (CNN) model at 97.945%.

Along with the same lines, Chen et al. [42] proposed an ML model for the intelligent detection of malicious URLs. They collected their dataset from Alexa [22], urlquery.net, urlscan.io [43], and GitHub [44]. Their study provided 41 lexical, net-work-based, and content-based features. The 17 most significant features were identified using analysis of variance (ANOVA) and the extreme gradient boosting algorithm (XGBoost). They concluded that the XGBoost classifier had the best result with 99.98% detection accuracy.

A study conducted by Vundavalli et al. [45] aimed to distinguish between benign and malicious websites They obtained their dataset from the Kaggle website [46]. The best result was achieved by naive bayes (NB) with an accuracy of 91%.

Additionally, Crisan et al. [47] proposed a method with the goal of using a combination of word embeddings and network-based features that considered specific methods of addressing the class imbalance. Their dataset was provided by a security company. The classifier with the best performance was a multilayer perceptron (MLP), with an accuracy of 95.81%.

2) LEXICAL AND CONTENT-BASED FEATURES STUDIES

Cao et al. [48] proposed a model to detect malicious URLs in online social networks (OSNs) using seven lexical and content-based features. They collected the original messages from the largest OSN in China, Sina Weibo. The bayesian network (BN) model achieved the best results with an accuracy of 84.74%. The limitations of this study included the lack of an expanded dataset, the collection of big data being a challenge for data mining, and the need for the evaluation to be more comprehensively compared with existing studies.

Humam et al. [49] evaluated various methods and offered rules-based applications for efficient phishing detection. The authors of this study built their dataset, and the detection methods were based on 13 lexical and content-based features. The experimental results showed that the decision tree (DT) had the highest accuracy, at 96.8%.

Similarly, Rao et al. [50] proposed a classification model based on lexical and content features to overcome the disadvantages of current anti-phishing techniques. Their dataset consisted of the Alexa PageRank system [22] and PhishTank [14]. Principal component analysis RF (PCA-RF) performed the best out of the oblique RF methods, with an accuracy of 99.55%.

A study conducted by Adewole et al. [51] proposed a hybrid rule induction algorithm capable of separating phishing websites from legitimate ones. The hybrid algorithm uses the strengths of both the rule induction algorithm (JRip) and the projective adaptive resonance theory (PART) algorithm to produce rule sets. Their dataset was collected from PhishTank [14], Yahoo, Alexa [16], CommonCrawl [52], and OpenPhish [36]. The total of extracted lexical, network-based, and content-based features was 40, with the proposed system returning the highest accuracy of 99.08%.

Kumi et al. [53] proposed a malicious URL detection method that uses a classification based -on -association (CBA) algorithm. They collected their dataset by crawling Alexa's top 500 sites [22], OpenPhish [36], VxVault [54], and URLhaus [55] and used 11 lexical and content-based features. Their model achieved an accuracy of 95.83%.

Liu et al. [56] designed a web spam detection method by extracting novel feature sets. They built their method based on the WEBSPAM-UK2007 dataset [57] and the UK-2011 [58]. In addition, they selected 28 content-based and lexical features. The highest accuracy was achieved by the RF at 93%.

3) LEXICAL AND NETWORK-BASED FEATURES STUDIES

Manjeri et al. [59], proposed a model to classify a URL by handling class imbalance using a public dataset [60]. The RF algorithm achieved the best accuracy at 96%.

Differently, a study conducted by Vanhoenshoven et al. [61] developed a model to detect malicious URLs and obtained better accuracy with the same classifier of 98.26%. The dataset they used was adopted from the one presented by Ma et al. [62]. The URLs were obtained from a large webmail provider and Yahoo's directory listing. The extracted lexical and network-based features were 3.2 million for these studies.

Another study conducted by Rakotoasimbahoaka et al. [63] aimed to solve the over-fitting problem of the combination of ML and DL by using different laws in a majority voting system. The datasets used were from OpenPhish [36]. They extracted 12 lexical and network-based features. They found that the majority vote method solved the over-fit problem in a combination model (RF–CNN–LSTM), which reached 93% accuracy using the second dataset.

A study conducted by Rao et al. [64] developed an ML model for classifying URLs using a dataset collected from Kaggle [60]. They used 19 lexical and network-based features. Finally, they implemented the system using XGBoost, which achieved an accuracy of 96.8%.

Rakotoasimbahoaka et al. [65] proposed a hybrid approach based on ML and DL methods. They used datasets from Kaggle and a combination of lexical and net-work-based features to get the best prediction. In the final experiment, they found that their proposed model CNN–LSTM–RF (96%) did not perform as well as CNN–LSTM (99%), but it detected URLs better.

In addition, a study conducted by Chiramdasu et al. [66] applied an ML approach to identify malicious URLs. The ML model was implemented using Logistic Regression (LR) with a dataset compiled from PhishTank [14], Kaggle, and GitHub public repositories [44]. To classify the URLs, network-based and lexical features were used, and the KNN model achieved the best accuracy with 93%.

Shi et al. [67] proposed an approach to detect malware domain names using Extreme Learning Machine (ELM). Their dataset was collected from DNS queries in the Network and Information Center of Shanghai Jiaotong University. In addition, they selected nine lexical and network-based features, and their detection method had a high detection rate with an accuracy of more than 95%.

Furthermore, Parekh et al. [68] introduced a model to detect phishing websites using URL detection. The dataset used was gathered from PhishTank [14]. The best accuracy came with the RF model, which reached around 95%.

Butnaru et al. [69] achieved a better result with an accuracy of 98.86% with their development of a phishing detection engine based on an ML model using nine features. Their dataset was formed from PhishTank [14] and Kaggle [70].

However, Shantanu et al. [71] achieved better accuracy of 99.7% with the same classifier in their comparison of the efficiency of several ML classifiers at detecting malicious URLs using 14 features. The dataset used was from the Kaggle repository [70]. All three studies extracted two types of features: network-based and lexical.

Another study that proposed an approach to detect malicious URLs was conducted by Astorino et al. [72]. They have used two datasets, which are PhishTank [14] and the second dataset from DMOZ Open Directory [73]. They selected seven lexical and network-based features. They obtained the best accuracy (86.3%) with a spherical separation methodology.

Another study proposes a new model by Peng et al. [74]. Their model was based on the attention mechanism (JCLA) for detecting malicious URLs. The dataset was from PhishTank [14], and the URLs were identified using 98 lexical and network-based features with the SoftMax classifier. The JCLA achieved an accuracy of 98.26%.

Wadas [75] presented a model to detect phishing URLs using ML techniques. The datasets used to train the model are from PhishTank [14], and another dataset was adapted from the author's previous work [76]. Lexical and network-based features are extracted in this model by a total of 14 features. The NN achieved the best results with an accuracy of 78.4%.

Sadique et al. [77] developed a framework for detecting phishing URLs with a dataset built from PhishTank [14]. They calculated the cost for each URL feature used. They noticed that lexical features require less time to extract than network-based features. As a result, they attempted to categorize URLs using the less expensive feature sets first before obtaining the more expensive ones. The experimental results showed that RF outperformed all other ML algorithms in terms of accuracy and time duration, with a score of 90.51%.

However, Patgiri et al. [78] proposed a model to detect malicious URLs that obtained better accuracy (93.30%) with the same classifier. They divided the dataset into training and test data in 60:40, 70:30, and 80:20 ratios. They calculated the accuracy for several iterations for each split ratio. As a result, they concluded that the 80:20 split ratio was the best split.

In contrast, a study conducted by Chiramdasu et al. [79] using the same classifier detected malicious URLs with an ML technique using 13 features and achieved an accuracy of 99.61%. All three studies extracted lexical and network-based features of the URLs.

Prieto et al. [80] presented a novel knowledge-based system called domains classifier based on risky websites (DOCRIW). Five network-based features and one lexical feature were selected. The LR achieved the best accuracy of 89%. The DOCRIW framework had some limitations, such as an insufficient number of features and a limited data sample size.

Another study proposing a Chrome extension that acts as middleware between users and malicious websites was published by Desai et al. [81]. The dataset was obtained from the UCI–ML Repository [82], and 22 lexical and network-based features were utilized. The experimental results showed that the RF algorithm returned the best accuracy (96.11%).

Akour et al. [83] investigated the effectiveness of ML for phishing detection. They used a dataset proposed by Vrbancic et al. [84] that contained 111 lexical and net-work-based

features. Ultimately, SVM was the best performing model and it achieved an accuracy of 96.30%.

Along with the same line, He et al. [85] proposed a feature selection method based on RF. The dataset collected from Alexa, MalwareDomainList [86], OpenPhish [36], Cybercrime-tracker [87], and 360.com [88]. There were originally 28 lexical and net-work-based features, and after the feature selection process, they became 18 in total. The best results were achieved by RF with an accuracy of 90.81%.

Ozcan et al. [89] proposed hybrid DL models that were combinations of deep NN (DNN)–LSTM and DNN–Bidirectional LSTM (BiLSTM). They used two datasets from Ebbu2017 [90] and PhishTank [14], along with a dataset from PhishStorm [91]. They have extracted the network-based and lexical features. The highest accuracy was achieved by DNN–BiLSTM, with an accuracy of 99.21% using the second dataset.

Lee et al. [92] conducted a study to assess the efficiency of the ML approach in detecting and identifying malicious and benign URLs. They used a public dataset from Kaggle that contains malicious and benign URLs. For classification model construction, they used nine network-based and lexical URL features. They employed features optimization techniques to select relevant URL features by utilizing a bio-inspired algorithm, which reduces the time for training and testing and simplifies the malicious URL detection system. Ultimately, both the NB and SVM models presented a performance of 99% accuracy, which was better than the other classifiers.

4) LEXICAL STUDIES

Raja et al. [93] proposed a method to detect malicious URLs. In order to detect the malicious URLs, they extracted 27 lexical features, but only utilized 20 features that reduce execution time and storage requirements. The study used the university of new brunswick (UNB) dataset [94]. The classifier achieved the best result with an accuracy of 99%.

Another study conducted by Vanitha N et al. [95] had the goal of allowing computers to learn independently, without human intervention or support, and consequently regulate actions. The dataset was collected from GitHub [96]. They considered lexical and other features. The websites were classified as malicious or benevolent [96], and the best result was achieved by LR with an accuracy of 98.42%.

Aalla et al. [2] proposed a model that detects malicious URLs based on comparing the results between two algorithms using LR and DT. They used a dataset that labelled URLs as legitimate or malicious. The LR model achieved the best result with an accuracy of 97.5%.

A study conducted by Ateeq et al. [97] had the goal of introducing a method to classify URLs according to their type using NN. The dataset used in this study was CICANDMAL2017 [98], and they extracted eight lexical features from URLs. They used a feedforward NN (FFNN), which falls under DL algorithms, with multiple hidden layers to detect the URL type. The NN was able to successfully detect 98.48% of the URLs.

Another study using lexical features was conducted by Shivangi et al. [99], who proposed a tool deployed as a Chrome extension using DL techniques. The authors collected several URLs from various sources by web scraping. The dataset was obtained from search engines, PhishTank [14], and CommonCrawl [52]. Finally, the LSTM model achieved the best results with an accuracy of 96.89%.

Likewise, a study that used lexical features only was conducted by Pingle et al. [100]. Their goal was to provide a structure for detecting a harmful web page, and they found that the best classifier was ID3.

Lakshmanarao et al. [101] proposed a model that helps detect malicious websites using lexical features. The dataset used was from Kaggle [102]. The best classifier was the RF with an addition to the hashing vectorizer (HV) technique, which achieved the highest accuracy of 97.5%.

Khan et al. [103] presented a model that used a majority voting classifier to combine numerous ML methods. The datasets used were obtained from UNB [94] and Kaggle [104]. They extracted 47 lexical features with the help of feature scoring techniques to identify the most frequent significant features in both datasets. The voting classifier achieved the highest accuracy of 99.72%.

Another study that introduced a phishing detection technique was conducted by Abutaha et al. [105] using a dataset published by another author [106]. The dataset was processed to produce 22 lexical features, and the best results were from SVM, with an accuracy of 99.896%.

Zhao et al. [107] focused on using ML techniques for multi-classification of malicious URLs. They used part of the dataset from [108] and [109], which were both derived from a Chinese Internet security company. The gated recurrent unit (GRU) NN model outperformed the RF model with an accuracy of 98.5%.

In addition, Hai and Hwang [110] presented a solution based on natural language processing (NLP) techniques to classify URLs as either benign or malicious. Their dataset was from DMOZ [32], MalwareDomainList [21], Malc0de [111], and CleanMX [112] Extracting lexical features of the URL. The best accuracy (97.1%) was achieved by SVM.

Another study focused on building an efficient and fast phishing URL detection approach was conducted by Banik et al. [113]. The used dataset was collected from PhishTank [14] and the DMOZ directory [114]. The performance was evaluated with different sizes of datasets using different numbers of features. A total of 18 lexical features were extracted from the URLs and the 15 most frequently contributed features were selected. The proposed system that used the SVM model was able to detect phishing websites with an accuracy of 96.35%.

Sameen et al. [115] designed an ensemble ML-based system called PhishHaven to detect both AI-generated and human-crafted phishing URLs. They classified the URLs as phishing and normal using 17 lexical features. The dataset was collected from Alexa [16], PhishTank [14], and DeepPhish

[116], and the results showed that PhishHaven was able to achieve 98% accuracy.

A study comparing the performance of traditional ML algorithms with popular DL framework models was conducted by Johnson et al. [11]. Two experiments were conducted using the ISCX-URL-2016 dataset from UNB [94] containing five URL classes: benign, defacement, malware, phishing, and spam. In order to classify the URLs, 78 lexical features were used. The best results were achieved by RF with an accuracy of 96.99%.

A study by Liang et al. [117 proposed an algorithm based on deep bidirectional long short-term memory (DBLSTM) the researchers used open datasets from 360 NetLab [118] and Alexa [119]. Lexical features of URLs were chosen due to the simplicity of analysis and widespread application to any kind of domain generation algorithm (DGA) family. The precision of the proposed DBLSTM algorithm remained high at 93–95%, while that of conventional models such as LR and SVM dropped significantly, to lower than 71–73%.

In addition, a study by Joshi et a. . [120] proposed a static lexical feature-based RF classification approach to classifying malicious and benign URLs they used a dataset from various sources, including OpenPhish [36], Alexa whitelists [16], and internal FireEye. After they analysed several URLs, they found 23 different lexical features that could be used to classify malicious and benign URLs. Ultimately, the RF model was the best choice for classification, with the best accuracy (92%) of the compared models.

One study conducted by Ispahany et al. [121 proposed an ML classification technique for detecting malicious URLs due to the COVID-19 pandemic using five lexical features. The used dataset was collected from DomainTools [122] and WhoisDS [123]. Their model achieved an accuracy of 99.2%. In the future, they plan to investigate the incongruence of entropy.

A study by Afzal et al. [124] introduced a hybrid DL approach named URLdeepDetect for time-of-click URL analysis and classification to detect malicious URLs using lexical features. This study used a dataset from PhishTank[14] and Kaggle[70]. The k-means model achieved the best results with an accuracy of 99.7%.

Another study conducted by Zeng [125] used 26 lexical features to detect malicious URLs in email content. The dataset used was from PhishTank [14] and DMOZ [32], and the experimental results showed that gradient boosting DT (GBDT) outperformed all other classifiers, achieving an accuracy of 90.71%.

Gupta et al. [126] developed an ML-based phishing detection system to help users check the legitimacy and maliciousness of a URL within a minimal time frame. This study used a dataset from the University of Canada Brunswick from UNB [94], and nine lexical features. They achieved the best accuracy of 99.57% with the RF algorithm.

Another study, which was conducted by Banik [127], developed an ML-based phishing URL detection system using lexical features of URLs. The dataset was collected from PhishTank[14], the DMOZ directory [114], a dataset proposed by Chiew et al.[128], and a dataset collected by the authors [129]. A total of 17 lexical features were extracted from URLs. They achieved the best accuracy of 98.57% with the RF algorithm.

Sahingoz et al. [130] proposed a real-time anti-phishing system using lexical features. The dataset was collected from PhishTank [14] and Yandex Search [131]. The RF algorithm using only NLP-based features gave the best performance with an accuracy of 97.98%.

Another study, conducted by Bahnsen et al.[132], focused on the classification of sites as legitimate or phishing using ML techniques. The dataset used was extracted from PhishTank [14] and Common Crawl [52], and 14 features were selected based on the URLs lexical and statistical analyses. The LSTM model achieved the best results with an accuracy of 98.7%.

Wei et al. [133] presented a method of detecting malicious URL addresses based on only URL lexical features using a DNN with convolutional layers. The dataset was collected from PhishTank [14], Alexa[16], and CommonCrawl [52]. Their model achieved an accuracy of 99.98%.

Yuan et al. [84] proposed a methodology that makes use of embedded representations of characters in URLs to detect phishing web pages. The character embedding achieved by the word2vec model does not depend on any network load or external knowledge. However, it does depend on lexical features (character embedding). The dataset used was collected from Alexa[119], the technical challenge of network security[134], PhishTank [14], and Reasonable Anti-phishing[135]. The best performance was achieved by XGBoost with an accuracy of 99.69%.

Additionally, Yang et al. [136] proposed an integrated phishing website detection method based on RF and CNN. They used two datasets: the first dataset (DS1) was compiled from PhishTank[14] and Alexa[16]. The second dataset (DS2) was a bench-mark dataset used by Sahingoz et al. [130] from PhishTank[14] and Yandex[137]. They extracted lexical features (character embedding features) using the CNN model and classified multilevel features using RF classifiers. Their model achieved an accuracy of 99.35% using DS1.

Yuan et al. [138] proposed a model that is based on the attention mechanism, bi-directional independent recurrent neural network (Bi-IndRNN), and CapsNet that when combined formed a joint NN algorithm model for detecting malicious URLs. The dataset was collected from the Alexa website [22], hpHosts [139], and PhishTank [14]. The extracted features were lexical and a texture fingerprint feature that converts the URLs into grayscale images. The proposed model achieved an accuracy of 99.78%.

5) CONTENT-BASED FEATURES STUDIES

Altay et al. [140] proposed classifying web pages using supervised ML techniques and a dataset collected from PhishTank [14] and Alexa [16]. The data were extracted from

web pages using a keyword density extractor library designed by Comodo Group [141], and 8,000 content features were extracted. The achieved accuracy of 98.24% with SVM-Radial basis function (SVM-RBF).

McGahagan et al. [142] proposed assessing whether additional webpage features would improve the detection of malicious websites. They collected their dataset from the Cisco Talos Intelligence Group [143] and Alexa list [22] and selected 26 content-based features. The RF model achieved the best accuracy of 91.36% in the case of no sampling.

In addition, Jain et al.[144] provided a novel approach for identifying phishing threats by examining hyperlinks in the HTML source code of a website. They collected the dataset from PhishTank [14], Alexa top websites [22], Stuffgate Free Online Website Analyzer [145], and the online payment service providers list. Their proposed approach combined a variety of unique remarkable hyperlink-specific features to detect phishing attacks and divided the hyperlink-specific features into 12 content-based feature categories. The best result was achieved by the LR model with an accuracy of 98.42%.

B. ARABIC STUDIES

Many studies have been conducted to investigate different features and algorithms to detect phishing attacks on Arabic content websites. This section presents these studies and categorizes them into two sections. The first section presents studies that focused on lexical and content-based features, and the second section presents studies focused on content-based features.

1) LEXICAL AND CONTENT-BASED FEATURES STUDIES

Al-Kabi et al. [146] proposed an approach to detecting Arabic spammed web pages using content-based analysis. They built their dataset using a crawler developed by Alsmadi [147] and selected seven lexical and content-based features. The results showed the DT algorithm was the best, with an accuracy of 99.521%.

Another research focused on web spam detection was conducted by Al-Kabi et al.[148] proposed an integrated online Arabic web spam detection system (OLAWSDS) that filters malicious pages from search engines. They extracted 18 lexical and content features. The best results achieved an accuracy of 99% using the trust rank model.

In addition, EL-Mohdy et al. [149] proposed web spam detection based on web mining. The spam web pages were collected manually using search engines with a spamming query such as pages that support terrorism from Egypt's blocked websites list [150]. The non-spam web pages were collected from trusted sites such as governmental and news sites. In addition, they selected one lexical feature and three content-based features. The DT classifier achieved an accuracy of 97%.

2) CONTENT-BASED FEATURES STUDIES

The study conducted by Alsaleh et al. [10] showed how ineffective Google's anti-spamming methods are against web spam pages that contain non-English content. It provided a solution in the form of a browser anti-spam plug-in detecting Arabic spam pages. The dataset was collected by the authors themselves, and they selected seven content-based features. Also, they tested four ML algorithms by using multiple variations to build their classifier. The results were that the performance of the random forest DT (RFT-S) showed the best detection rate, which was 87.13%.

Similarly, Wahsheh et al.[151] proposed a system to classify URLs as spam or not spam. The goal of the study was to build the first Arabic content or link web spam detection system using the rules of DT. The proposed system helps to clean a search engine results page (SERP) of all URLs referring to Arabic spam web pages. The proposed model achieved 93.1034% accuracy for Arabic links using 15 content-based features.

Another study, published by Al-Twairesh et al. [152], aimed to analyse the content of Saudi tweets to detect spam by developing both a rule-based approach and a supervised learning approach. They used the Twitter search application programming interface (API) to collect the dataset for spam and non-spam tweets. The NB classifier gave the best results by stemming 91.6% using four content features.

Likewise, Alorini [153] proposed discovering Arabic spam on Twitter using ML. The dataset was collected from Twitter using Twitter stream API and the Tweepy package from Python and then translated into English with the help of Arab annotators. Three content-based features were selected in this study. The highest accuracy of 91% was achieved by Bayesian reasoning (BR).

Additionally, Alkhair et al. [154] focused on investigating fake news content in the Arabic world through the information posted on YouTube. They collected comments that were classified as rumor or non-rumor using the YouTube API. The achieved performance varied depending on the rumor topic and the classifier used. Overall, for the dataset used, the best classifier was the SVM, which reached an accuracy of 95.35%.

Wahsheh et al. [155] proposed an approach to detect link-based spamming techniques used in Arabic spam web pages. The dataset was collected using Web Link Validator [156] to analyse the web pages by finding broken links, checking the HTML code's accuracy, and selecting six content-based features. The DT yielded the highest accuracy of 91.4706%.

Mataoui et al.[157] proposed a new supervised spam detection approach by defining a set of features in the Arabic language. The dataset was extracted from Facebook. In the pre-processing step, they extracted tokens using standard NLP techniques, such as tokenization, normalization, stop-word removal, and stemming. The normalization stage in the Arabic language serves to convert each letter to its prescribed standard form (for example, "ا، أ، إ، آ، ا" are multiple forms for

the letter "ا" [alif]). The J48 model achieved the best results with an accuracy of 91.73%.

Another study that used content-based features of Arabic tweets was proposed by Alharbi et al. [158]. They focused on classifying rogue and spam content in Arabic tweets using ML algorithms. They collected the dataset from spamming Twitter accounts. The 47 generated features were analysed, and the best features were selected. The performance results of the study showed that the RF classification algorithm with 16 features performed best, achieving accuracy rates greater than 90%.

Najadat et al. [159] proposed a keyword-based method to detect Arabic spam reviews. The dataset was extracted from different sections of Facebook pages using the Netvizz application [160]. The Facebook comments were classified based on content-based features, such as the keywords extracted from them. The best results were achieved by the DT model, with an accuracy of 92.63%.

In addition, Mubarak et al. [161] proposed a model to detect Arabic spam tweets and identified different properties of Spam and Ham tweets. They built their own dataset from Twitter, and they selected four content-based features. The highest result was achieved by the Arabic bidirectional encoder representations from transformers (Ara-BERT), with an accuracy of 99.7%.

Alsulami et al. [162] proposed a personalized filtering model they called the SentiFilter that aimed to provide each user with a personalized level of protection against what the user perceives as unwanted content. The dataset was collected from Twitter. The best results were achieved by the SVM classifier, with an average accuracy of 90.89%.

Wahsheh et al.[163] proposed Arabic opinions spam detection system (SPAR). The goal was to detect spam opinions in the Yahoo!–Maktoob social network and categorize them as spam or non-spam opinions based on many features. A dataset of opinions (reviews) from Yahoo!–Maktoob News was collected and analysed by the authors. Each data gathering opinion must be pre-processed using the following procedures: 1) Delete the non-Arabic text. 2) Delete the punctuation. 3) Normalize the similar Arabic letter. 4) Tokenize the Arabic opinion. They used SVM to evaluate the proposed SPAR system, and it achieved an accuracy of 97.5073%.

Another study focused on Arabic tweets for the detection of suspicious messages was conducted by AlGhamdi et al.[164]. The goal was to develop a system to detect suspicious messages written in the Arabic language. They used the Twitter streaming API to get Arabic tweet data. The SVM model achieved the best results with an accuracy of 86.72%

V. DISCUSSION AND ANALYSIS

This section summarizes the reviewed papers in terms of publication year, URL classification, dataset source and size, classifiers, and the highest results obtained, as previewed in Table 1. In the English studies, the most frequently used features were lexical features, in 72 studies out of 91. That was followed by network-based features, which were used in 39 studies, while 26 studies used content-based features. We can conclude that the highest result was achieved by CNN with an accuracy of 99.98% [133] on a dataset size of 21,208 URLs. In contrast, the Arabic studies mostly used content-based features, in 16 studies, while three studies used the URL lexical features. In contrast, network-based features were not used in the Arabic content websites. We can conclude from the Arabic studies that the highest result, with an accuracy of 99.521%, was achieved by DT [146] on a dataset of 15,000 Arabic spam web pages.

The majority of English-based studies that used lexical, network-based, and content-based features achieved high accuracy that were greater than or equal to 95%. On the other hand, none of the Arabic-based studies used all three types of features together. Noteworthy, the English-based studies that used all three types of features and achieved the highest accuracy is the one conducted by Chen et al. [42] with an accuracy of 99.98%. This study showed that three network-based features represent the most important features which are the following:

1. Whether the domain country code is included in the top eleven common malicious country codes or not.
2. The interval between the domain update time and the current time.
3. The interval between the contract expiration of the domain and the current time.

Unfortunately, there are not many English-based studies that utilized these three features. Even more, there is no Arabic-based study that used any of the network-based features. Therefore, using the combination of the three types of features in an Arabic-based study could be a new promising research direction to explore.

Several studies combined two kinds of URL-based features: lexical-content-based and lexical-network-based features. The lexical-network-based features were the most used combination of URL-based features by various studies, including [71], which achieved the highest accuracy of 99.7% using the RF classifier. However, [50] achieved a greater accuracy of 99.55 % by combining lexical and content-based features and employing the PCA-RF classifier.

From our review, we found that the number of utilized lexical features only were in the range of 5 to 47 features. The lexical features are the most used type of features due to the following reasons:

1. The lexical features can be extracted without the need for additional services, tools, or an Internet connection.
2. Most of the outputs are numbers so they did not require any sort of encoding such as (URL length, number of special characters, etc).
3. Fast execution time.

Some of the popularly used features are URL length, length of the domain name, count of some symbols such as ('@','&', #,'/','','.'), and count of digits. Furthermore, the special characters mentioned are considered suspicious characters and they are highly present in the phishing URLs.

Moreover, the attackers tend to use long URLs to hide suspicious parts [110]. They also use the redirecting symbol "//" to allow the redirection of the websites containing the attack [110]. They may sometimes use some of the suspicious words within the URL such as the word tokens (e.g., sign in, confirm, free, etc.) [110].

Two years ago, the researchers focused on deploying a detector as quickly as possible to detect malicious URLs associated with a certain trend, such as the COVID-19 pandemic. During the COVID-19 pandemic, UW Medicine made extensive use of telemedicine capabilities to provide patients with virtual care [165]. Staff members noted a dramatic increase in phishing emails that enticed employees to click on malicious links and download malware during this period [121]. Although the lexical technique is fast, it might not be sufficient to guarantee complete security if attackers attempt to hide dangerous information behind normal URLs using benign tokens.

The content features without any additional features are the least used features in English and Arabic studies. Some of the widely used content features are, the number of words, the maximum number of words within certain HTML tags (<bod>, <head>, etc), and the count of certain HTML tags such as (<met>, <img>). The meta tags are typically used to specify the page description, keywords, and author of the document. So, the hackers utilize them to enhance the page rank by utilizing keyword stuffing. Usually, phishing sites contain more images than benign ones. To extract the content features, the researchers would need to consider the HTML content of webpages, and JavaScript (<iframe> method, etc.). The content requires a set of pre-processing steps including the removal of stop words and some special characters. Moreover, some languages need the removal of "Tashkeel" and "Tatweel" like the Arabic language. Some studies that are only concerned about the content features may face different challenges such as customizing the detection model to handle different languages. However, the selection of the type of features usually relies on the URL dataset or the attack type, such as spam, phishing, drive-by-downloads, and malware.

In terms of the classification algorithms, the following set of algorithms achieved the best performance in terms of accuracy of 99% and above: CNN, XGBoost, LSTM, SVM, CW, Majority Voting Classifier, RF, K-means, Ara-means, DT, and NB.

Even though CNN, XGB, and LSTM achieved the highest results close to 100%, they are rarely used. The major disadvantage of using XGB is related to being very sensitive to outliers and is hardly scalable [166]. CNN is mostly suitable for image data. Additionally, the LSTM takes a longer time, requires more memory to train, and is easy to overfit.
On the other hand, the SVM, RF, DT, NB, and LR are the mostly used algorithms that achieved good performance with an accuracy of 98.42% and above. It should be noted that all of these algorithms are ML classifiers. The ML classifiers work well on small and large datasets whereas the DL classifiers work well with large datasets [167].

It is important to note that the Ensemble technique, which combines a set of algorithms, provides high accuracy of more than 90%, as shown in Table 3. In general, the ensemble method outperforms the individual models in terms of accuracy [168].
Furthermore, the algorithms that have low performance are BN, NN, and DBN achieving an accuracy of lower than 90%. The major drawback of the NN and DBN is the need for a large amount of training data. Besides, the training process of the NN is the focal point of deciding the correct prediction of data patterns [169].
In addition, there is no good or bad algorithm due to many factors such as how clean and good the pattern of a dataset is, the size of the dataset, and the number of features.

In terms of the dataset, a total of 45 different dataset sources were used. The most common dataset source is PhishTank [14], which is available in multiple formats and is updated hourly. Datasets built by the study's authors were the second most datasets used in the studies and were collected by using crawling tools, special APIs, or manually. In the studies that were based on Arabic content websites, all authors used their built dataset, since there is a lack of datasets for Arabic content websites. The third most dataset source is Alexa [16]. There are datasets sources that were used in more than one study as well, such as Kaggle, OpenPhish [36], and CommonCrawl [52]. However, compared to PhishTank [14] and Alexa [16] they are considered less popular. Some dataset sources were not used frequently such as Ebbu2017 [90], CleanMX [112], DMOZ Open Directory [73], and WEBSPAM-UK2007 dataset [57].

Table 1 **English content websites explored studies**

Reference	Year	URL classification	Dataset		Classifier	Results
			source	size		
[2]	2022	Malicious or legitimate	Not mentioned	420,000 URLs are legitimate and malicious URLs.	LR and DT	LR - accuracy: 97.5%
[93]	2021	Benign, phishing,	"ISCX-URL2016" from UNB [94]	68,851	RF, LR, KNN, NB, and Support Vector Classification(SVC).	RF - accuracy: 99%.

		malware, or spam				
[97]	2021	Benign, defacement, malware, phishing, or spam	CICANDMAL2017 [98]	500	FFNN	Accuracy:98.48%
[124]	2021	Malicious or benign	Kaggle[70] and PhishTank[14]	450,176	RF, MLP, and NB, LSTM, and k–means clustering.	K-means clustering - accuracy : 99.7%
[126]	2021	Phishing or benign	Repository of the University of Canada Brunswick from UNB [94]	19,964	RF, KNN, LR, and SVM	RF - accuracy: 99.57%
[20]	2021	Malicious or benign	PhishTank[14],mal ware domain list [21] and Alexa[22]	66,017	(CapsNet and IndRNN)	Accuracy: 99.78%.
[121]	2021	Malicious or benign	DomainTools [122] and WhoisDS [123]	7849	(SVM, KNN, NB)	Accuracy: 99.2% by using all models
[53]	2021	Malicious or benign	Alexa's top 500 sites [16], OpenPhish[36], VxVault [54] and URLhaus [55]	1200	CBA	95.83%
[66]	2021	Safe or malicious	PhishTank [14], Kaggle, and GitHub public repositories [44]	more than 32,000 URLs	LR, SVM, KNN, and linear discriminant algorithm (LDA).	KNN - accuracy: 93%
[101]	2021	Phishing or benign	Kaggle [102]	5,49,346,1,56 ,422 phishing,3,92, 924 benign URLs	KNN, DT, RF, and LR	RF - accuracy: 97.5%
[69]	2021	Phishing or benign	Kaggle [70], PhishTank[14]	40,000 benign, 60,315 phishing URLs	NB, DT, RF, SVM, and MLP	RF - accuracy: 98.86%
[71]	2021	Malicious or benign	Kaggle [70]	450,000 URLs	LR, Stochastic Gradient Descent (SGD),NB,KNN,DT, RF, and SVM	RF - accuracy: 99.7%
[138]	2021	Malicious or benign	Alexa [22],Hphosts [139] and PhishTank [14]	32,378 benign URLs, malicious URLs 33,549	Joint NN- (Bi–IndRNN) - (CapsNet)	Joint NN- (Bi–IndRNN) - (CapsNet): accuracy: 99.89%
[85]	2021	Malicious or benign	Malwaredomainlist [86],OpenPhish [36], Cybercrime-tracker[87], and 360.com[88]	400,000 URLs	Gradient Boosting (GB), adaptive boosting (AdaBoost), RF, SVM, LR, Gaussian Naive Bayes (GNB), and KNN	RF- AUC: 90.81%
[105]	2021	Malicious or benign	another author[106]	1,056,937 labeled URLs	RF, GB, SVM, and NN	SVM - accuracy : 99.89%
[41]	2021	Phishing or benign	PhishTank [14] and Alexa[16].	40,000	DT, SVM, KNN , MLP and CNN	CNN - accuracy: 97.94%.
[136]	2021	Phishing or legitimate	PhishTank[14], Yandex[137], and Alexa[16].	DS1:47,210 DS2 : 83,857	CNN and RF	(CNN – RF) - accuracy: 99.35%
[89]	2021	Phishing or legitimate	PhishTank[14], PhishStorm[91], and Ebbu2017[90].	DS1= 73,575 and DS2= 26,000	NB, KNN, AdaBoost, DT, Ridge regression, Least Absolute Shrinkage, and Selection Operator (LASSO), LightGBM, XGBoost, RF, DNN, CNN, Recurrent Neural Networks (RNN), LSTM,	(DNN+BiLSTM) - accuracy: 99.21%

					BiLSTM, DNN-LSTM, and DNN-BiLSTM	
[49]	2021	Phishing or legitimate	Private dataset	500 phishing and 500 legitimate web pages.	Rule–based, SVM, DT, and GNB.	DT - accuracy: 96.8%.
[80]	2021	Risky or non-risky	Collected by authors	1,500 domains	LR, RF, Extremely Randomized Trees (ERT), AdaBoost, GB, SVM, KNN, NB, and LDA	LR - accuracy: 89%
[83]	2021	Phishing or legitimate	Vrbancic et al.[84]	two versions containing 58,645 and 88,647 URLs	KNN, SVM, NB, and LR	SVM - accuracy: 96.30%
[79]	2021	Malicious or legitimate			RF, ID3, MLP, and NB	RF - accuracy: 99.61%
[100]	2020	Malicious or authentic			SVM, DT, and ID3	The best classifier was ID3
[45]	2020	Malicious or benign	Kaggle malicious dataset [46]		LR, NB, and CNN	NB - accuracy: 91%
[23]	2020	Malicious or benign	PhishTank[14]	1,089,012	(DBN–SVM), SVM, CNN, and LR	(DBN-SVM) – accuracy: 99.96%
[127]	2020	Phishing or benign	PhishTank[14], DMOZ directory [114], dataset proposed by Chiew et al.[128], and the dataset collected by the authors [129]	553,250	RF, DT, NB, and SVM.	RF - accuracy: 98.57%
[24]	2020	Phishing or benign	Kaggle [25]	11,055	SVM, NB RF, NN, BC , KNN,(NN –RF–BC), and (KNN +RF +BC)	(NN +RF +BC) - accuracy: 97.4%
[133]	2020	Phishing or benign	PhishTank[14], Alexa[16], CommonCrawl [52]	21,208	CNN	Accuracy: 99.98%
[17]	2020	Malicious or benign	They collected the dataset	470.000 URLs	SVM and RF	RF - accuracy 96.28%
[11]	2020	Benign, defacement, malware, phishing, or spam	ISCX-URL-2016 from UNB [94]	36,707 URLs.	RF, CART (DT), KNN, SVM, LR, LDA, AdaBoost, NB, Fast–AI, and Keras–TensorFlow	fast.ai - accuracy: 97.55%
[63]	2020	Malicious or benign	OpenPhish [36]	DS1:420,46 DS2:194, 798URLs.	RF, LSTM, and CNN	RF-CNN-LSTM-accuracy: 93 % by using all models
[47]	2020	Malicious or benign	Two private datasets	DS1: 500000 DS2:8 million	Cost–Sensitive NN, MLP, and Extra Trees (ET)	MLP - accuracy: 95.81%
[33]	2020	Legitimate or phishing	PhishTank [14]and Alexa [22]	5500 phishing sites and 5500 legitimate sites	SVM, Proximal Support Vector Machine (PSVM), and TWSVM	TWSVM-accuracy : 98.05%
[56]	2020	Spam or not spam	WEBSPAM-UK2007 [57] and UK-2011 [146]	5797 pages from WEBSPAM-UK2007 and 3766 pages from UK-2011	NB, LR, SVM, RF, CNN, RNN, and LSTM	RF - accuracy: 93%
[77]	2020	Phishing or benign	PhishTank [14]	60000 benign URLs and 38000 phishing URLs	KNN, AdaBoost, Gradient Boot (GDB), DT, RF, GNB, LDA, Quadratic Discriminant Analysis (QDA), SVC, and	RF - accuracy: 90.51 %

					Nu–Support Vector (NuSVC).	
[42]	2020	Malicious or benign	Alexa [22] for benign URLs and urlquery.net, urlscan.io [43], and GitHub [44] for malicious URLs.	26,054 URLs (13,027 benign and 13,027 malicious URLs).	KNN, DT, SVM, and XGBoost.	XGBoost - accuracy: 99.98%.
[26]	2020	Malicious or benign	Kaggle [27] and Urcuqui et. al [28] dataset	1781 malicious and benign URLs	RF, BN, J48, and KNN.	RF - accuracy: 95%.
[115]	2020	Phishing or normal	Alexa [16] for normal URLs, PhishTank [14] for simple phishing URLs, and DeepPhish [116] for AI-generated phishing URLs	50,000 normal URLs, 50,000 simple phishing URLs, and 50,000 AI-generated phishing URLs	AdaBoost, GB, DT, LR, SVM, and NN	Accuracy: 98%
[64]	2019	Malicious or benign	Kaggle repository of malicious and benign URLs [60]		XGBoost	XGBoost - accuracy: 96.8%
[95]	2019	Malicious or Benevolent	GitHub [96]		NB, RF, and LR	LR - accuracy: 98.42%.
[50]	2019	Phishing or benign	PhishTank[14] and Alexa[22]	3,526	J48 tree, RF, Sequential Minimal Optimization (SMO), LR, MLP, BN, SVM, and AdaBoostM1.	(PCA-RF) - accuracy: 99.55%.
[51]	2019	Phishing or benign	PhishTank[14]Yaho, Alexa[16] , CommonCrawl[52] and OpenPhish [36]	12,408	JRip, PART, and Hybrid rule–based (JRip + PART).	Hybrid rule-based - accuracy: 99.08%
[130]	2019	Phishing or legitimate	PhishTank[14] and Yandex Search[131].	73,575	NB, RF KNN, AdaBoost, K–star, SMO, and DT.	RF - accuracy: 97.98%
[140]	2019	Malicious or benign	PhishTank [14] and Alexa [16]	120773 websites	SVM, maximum entropy (MaxEnt), and ELM.	(RBF-SVM) accuracy: 98.24%
[142]	2019	Malicious or benign	Cisco Talos Intelligence Group [143] and the Alexa list [22]	40709 websites	AdaBoost, ET, RF, GB, BC, LR, KNN, and NN	RF - accuracy : 91.36%
[92]	2019	Malicious or benign	Kaggle		AdaBoost, SVM, KNN, NB, and RF	NB - accuracy: 99% SVM - accuracy: 99%
[75]	2019	Phishing or benign	PhishTank[14] , author [76]	10,480 phishing URLs and 11,000 benign URLs	DT, NN, and NB	NN - accuracy: 78.4%
[74]	2019	Malicious or benign	PhishTank [14]	7000 URLs	JCLA model	JCLA - accuracy: 98.26%
[65]	2019	Malicious or benign	Kaggle	DS1: 20,000, DSt2: 40,000 DS3: 224,480	(CNN–LSTM–RF), (RF–CNN–LSTM) and CNN_LSTM	CNN_LSTM accuracy: 99%
[117]	2019	Malicious or benign	360 NetLab [118] and Alexa [119].	1,145,093 malicious domains	LR, SVM, and DBLSTM	DBLSTM - accuracy: 95%
[120]	2019	Malicious or benign	Openphish [36], Alexa whitelists [16], and internal FireEye	approximatel y 5 million	RF, NB, SVM, AdaBoost, GB, and LR	RF - accuracy: 92%
[59]	2019	Malicious or benign	Malicious and Benign Webpages	1781 URLs	SVM, KNN, DT, NB, and RF	RF - accuracy: 96%

			Dataset from IEEEdataport [60]			
[103]	2019	Benign, spam, phishing, malware, defacement, or malicious.	UNB dataset, Kaggle [104]	UNB dataset 165,366 URLs and Kaggle 420,464 URLs	KNN, LR, SVM, AdaBoost, GB, RF, ET, and Voting	Majority Voting - accuracy: 99.72%
[107]	2019	Directory traversal, SQL injection, XSS injection, sensitive file attack, other attacks, or legitimate	Chinese Internet security company [108][109]	240,000 malicious URLs, and more than 150,000 legitimate URLs	GRU and RF	GRU - accuracy: 98.5%
[29]	2019	Phishing, spamming, malware, attack page , SQL injection, Gumblar, Fastflux, or denial of service	-	-	RF	Accuracy: 97%
[39]	2019	Phishing or legitimate	Phishtank [14] and DMOZ [40]	1,021,758 phishing URLs and 989,021 legitimate URLs.	AdaBoost, RF, GBDT, XGBoost, CNN–LSTM, and MFPD.	MFPD - accuracy: 98.99%.
[78]	2019	Malicious or benign	Collected by authors	Not mentioned	RF, SVM	RF - accuracy: 93.30%
[144]	2019	Phishing and non-Phishing	PhishTank [14], Alexa top websites [22], Stuffgate Free Online Website Analyzer [145], and online payment service providers list.	2544 phishing and non-phishing	SMO, NB, RF, SVM, AdaBoost, NN, C4.5, and LR.	LR - accuracy: 98.42 %.
[99]	2018	Malicious or non-malicious	search engines, PhishTank [14], and common crawl [52]	456,300	Artificial Neural Networks (ANN) and LSTM	LSTM - accuracy: 96.89.%
[125]	2018	Phishing or benign	PhishTank [14] and DMOZ [32]	30,000	LR, SVM, DT, RF, and GBDT.	GBDT - accuracy: 90.71%
[113]	2018	Phishing and non-phishing	PhishTank [14] and DMOZ directory[114].	32,652 URLs	SVM, NB, MLP, LR, and Decision Table	SVM - accuracy: 96.35%.
[68]	2018	Phishing or benign	PhishTank [14]	Not mentioned	RF	RF - accuracy: around 95%
[37]	2018	Benign, phishing, malware, or spam.	Alexa [22]. ,PhishTank[14], Malware Domain List[21], and jwSpamSpy[38][38]	49,935	OVA– SVM, OVO – SVM, and MC–CW	identification of attack types: 98.44%. In the detection of malicious URLs: 99.86% using the CW
[110]	2018	Malicious or benign	DMOZ [32] , Malware Domain List [21], Malc0de [111], and CleanMX [112]	150,396	SVM, LR, RF, KNN, and NB	SVM - accuracy: 97.1%
[84]	2018	Phishing or legitimate	Alexa[119], technical challenge of network security	1,172,577	GBDT, KNN, LR, RF, DT, and XGBoost	XGBoost - accuracy: 99.69%.

			[134], PhishTank[14], and Reasonable Antiphishing[135].			
[30]	2018	Benign, malicious, malware, phishing, or spamming and APT	Malware Domain List[21], Spambase Dataset[31] , Phishing Dataset[19], DMOZ[32] and Alexa[16].	27,700	DBN, ANN, SVM, and NB	DBN - accuracy: 75%.
[34]	2018	Malicious or benign	For malicious : MalwareURL, MalwareDomains (risk analytics now) [35], PhishTank [14], OpenPhish [36], and MalwareDomainList [21]. For benign URLs: Alexa [22], and DMOZ directory [32].	DS1:116,800 URLs, DS2:25,000 malicious and benign URLs.	RNN, Identity–RNN, LSTM, CNN, and (CNN–LSTM)	LSTM - accuracy: 99.96%
[132]	2017	Legitimate or phishing	PhishTank [14],and commoncrawl [52]	2,000,000	RF and LSTM	LSTM - accuracy: 98.7%.
[18]	2017	Phishing or benign	UCI - ML Repository [19]	11,055	ANN, KNN, SVM, C4.5 DT, RF, and Rotation Forest (RoF)	RF - accuracy: 97.36%
[67]	2017	Malware and benign	By the authors through DNS queries received by DNS servers in the Network and Information Center of Shanghai Jiaotong University	51011 domains	ELM	ELM - accuracy: 96.29%
[72]	2017	Malicious or benign	PhishTank [14] and DMOZ Open Directory [73]	DS1 = 580 URLs DS2 = 10952 URLs	Spherical separation	Accuracy: 86.3%
[81]	2017	Phishing or benign	the UCI - ML Repository [82].	11,055	KNN, SVM, and RF	RF - accuracy: 96.11%
[48]	2016	Malicious and legitimate	By authors	12,006	BN, J48, and RF	BN - accuracy of 84.74%
[61]	2016	Malicious or benign	Adopted from [62]	2.4 million URLs	NB, SVM, MLP, DT, RF and KNN	RF - accuracy: 98.26%
[15]	2012	Malicious or benign	Alexa[16]PhishTank [14]	10000 records	NB and Genetic Algorithm	Precision of 87% using two models

A. MACHINE LEARNING (ML) TECHNIQUES USED

The ML studies explored in this section are based on many aspects, such as the highest accuracy, considering the algorithm's name, the number of studies, and the algorithm's highest accuracy. Table 2 below shows the highest accuracy for each algorithm among all the studies that used this algorithm. It provides an overview of individual algorithms with the classification method and category of the algorithm. It is noteworthy that the highest accuracy (99.98%) was achieved by Wei et al. [133] and Chen et al. [42] using CNN and XGBoost, respectively. In addition, as demonstrated in Table 2, the SVM algorithm is considered one of the most frequently used ML algorithms in URL classification and was used in 47 studies. The SVM algorithm achieved an accuracy of 99.89%. However, the SVM algorithm cannot handle large or noisy datasets. Table 3 lists the combination algorithms, and Figure 2 shows the statistics of studies per algorithm.

Table 2 ML algorithms used in the reviewed papers

Algorithm	Classification method	Category	No. of Articles	Performance (Highest Accuracy)
CNN	Supervised	DL	7	99.98% by [133]
XGBoost	Supervised	Ensemble	5	99.98% by [42]
LSTM	Semi-supervised	DL	7	99.96% by [34]
SVM	Supervised	Regression	47	99.896% by [105]
CW	Supervised	Online learning algorithm	1	99.86% by [37]
Majority Voting Classifier	Supervised	Ensemble	1	99.72% by [103]
RF	Supervised	Ensemble	42	99.70% by [71]
K-means	Un-Supervised	Clustering	1	99.7% by [124]
AraBERT	Un-Supervised	DL	1	99.7%
DT	Supervised	Decision Tree	26	99.521% by [146]
NB	Supervised	Bayesian	32	99.00% by [92]
LR	Supervised	Decision Tree	28	98.42% by [95], [144]
MFPD	Supervised	DL	1	98.99%
GRU	Supervised	DL	1	98.5%
FFNN	Supervised	DL	1	98.48% by [97]
TWSVM	Semi-supervised	ML	1	98.05%
ELM	Semi-Supervised	ML	2	96.29% by [67]
CBA	Supervised	ML	1	95.83%
MLP	Supervised	DL	8	95.81% by [47]
DBLSTM	Semi-Supervised	DL	1	95%
KNN	Supervised	Instance-Based	29	93% by [66]
J48	Supervised	Decision Tree	4	91.73 % by [157]
BC	Supervised	Ensemble	3	91.05% by [142]
BR	Supervised	Bayesian	1	91%
GBDT	Supervised	ML	4	90.71% by [125]
BN	Supervised	Bayesian	4	84.74% by [48]
NN	Un-Supervised	ML	6	78.4% by [75]
DBN	Supervised	DL	1	75% by [30]

Table 3 Combined algorithms

Algorithm	No. of Articles	Performance (Highest Accuracy)
Joint NN- (Bi–IndRNN) - (CapsNet)	1	99.89% by [138]
IndRNN- CapsNet	1	99.78% by [20]
PCA-RF	1	99.55% by [50]
DNN + BiLSTM	1	99.21% by [89]
SVM, KNN, NB	1	99.2% by [121]
Hybrid Rule-based	1	99.08% by [51]
JRip + PART	1	99.08% by [51]
CNN-LSTM	4	99% by [65]
JCLA model	1	98.26% by [74]
SVM with an RBF kernel (SVM-rbf)	1	98.24% by [140]
RF-LSTM-CNN	2	93.59% by [63]
NB with Genetic Algorithm	1	precision of 87% by [15]

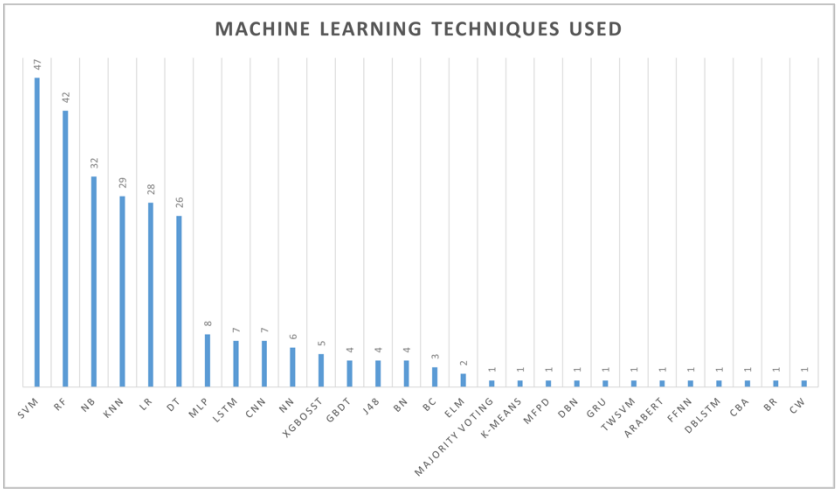

Figure 2. The most frequently used ML algorithms in the reviewed studies.

B. DATASETS USED

As mentioned in Section 4, the reviewed articles used datasets from different sources to train and test their detection models. Some of the datasets are open source, built by the study's authors, adopted from other authors, or a combination of those sources. However, the most common dataset sources are PhishTank [14] and Alexa [16]. PhishTank [14] was launched in 2006 by OpenDNS [171] and acquired by Cisco in 2015 [172]. PhishTank [14] is a free community site that enables anyone to submit, verify, track, and share phishing data. In contrast, Alexa [16] which was founded in 1996 by Brewster Kahle and Bruce Gilliat [173] acquired by Amazon in 1999 [174]. Alexa [16] provides up-to-date web global rankings, traffic data, and other information on over 30 million websites [175]. Also, Alexa is used for benign URLs because it is an analytical tool that lists the top-ranked URLs around the world or datasets collected by the study's authors. Most English-language content studies have used these two public datasets. In the same level of frequent use of the PhishTank dataset, the study authors used their own built dataset. In the Arabic-language content studies, all authors used their built dataset. Conversely, some sources were not often used despite representing further opportunities for research, including ArabicWeb16, which has 150 million Arabic web pages, making it the largest public Arabic web dataset. All the investigated studies had the same goal, which is to use these datasets to classify URLs. However, the authors differed in the next steps in terms of using the dataset with or without pre-processing, extracting more features, or further classifying the malicious URL as malware, phishing, or other. Figure 3 illustrates the investigated datasets along with the number of studies that used them.

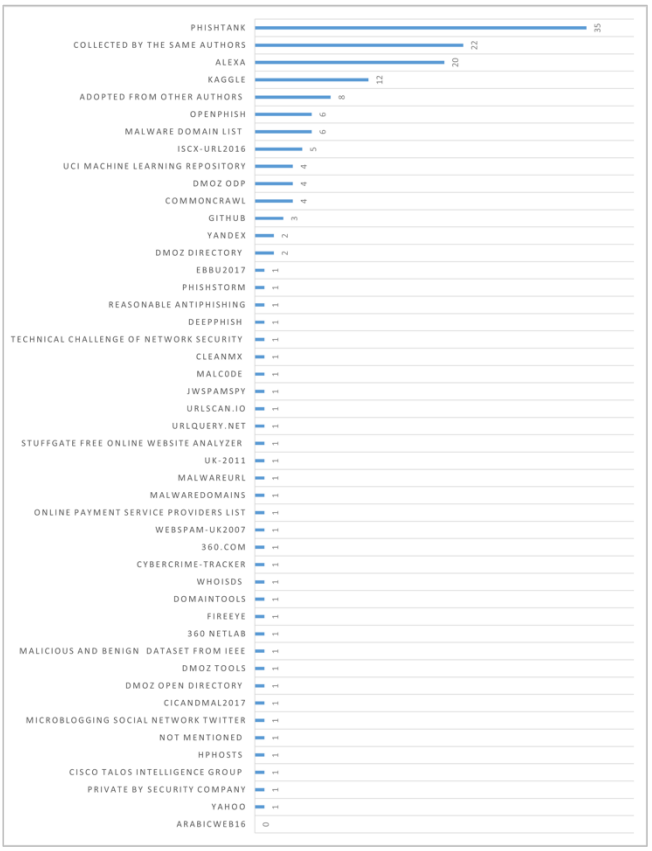

FIGURE 3. Datasets used in the reviewed studies.

VI. CHALLENGES AND RECOMMENDATIONS

Despite many significant improvements in malicious URL identification utilizing ML approaches over the previous decade, there are still many crucial and imperative unresolved problems and difficulties.

One of the main limitations of the reviewed papers was the data sample size[80],[162]. Therefore, we recommend evaluating and validating ML models for detecting malicious

URLs using enough samples with an acceptable ratio between the normal and malicious URLs. Balancing techniques can be used to enhance the quality of the detection rate while still taking into consideration enough samples in the dataset. On the other hand, big data collection is a challenge for data mining[48] due to the required processing computation and time. Overcoming this issue requires a scalable environment, such as cloud computing models or servers with enough computation power.

Moreover, there are other limitations, including the lack of analysis and detection of obfuscated JavaScript in web pages[37]; outlier values[75],[83]; number of features selected [74], [80]; and features effectiveness[74]. Issues related to the type and number of selected features can be resolved by applying selection techniques such as the following: (1) Filter feature selection methods that apply a statistical measure to rank the importance of the features. This includes the Chi-square test, information gain, and correlation coefficient scores. (2) Wrapper methods that deal with feature selection as a search problem followed by a searching technique such as best first search, random hill-climbing, or heuristic algorithms to evaluate a combination of features and rank features based on model accuracy. (3) Regularization methods that process feature selection as an optimization problem by applying regression techniques for minimizing the model coefficients by removing unrelated features.

The continuous change of inclusive features that differentiate between legitimate and suspicious URLs is also a major challenge that could be addressed in future research. A possible solution in this regard is investigating the applicability of Concept Drift detection techniques for enhancing the performance of intelligent phishing URLs. However, this requires incorporating a method for detecting the drift in concept in order to warn the model designer about the necessity of building a new model. Yet, building a new model should not mean ignoring the old model since the set of features that were inclusive at time t1 and become useless at time t2 might return to be inclusive at time t3. In fact, completely ignoring the old model might result in what is called catastrophic forgetting. Nevertheless, catastrophic forgetting can be addressed using different techniques including an ensemble approach by merging different models together where each model will produce a decision and then all decisions will be collected and processed to come up with a final decision based on the improved voting technique that considers the quality of each model in the ensemble.

Further, consideration must be given that the network traffic in a test environment and a real-world network are different. With the development of the Internet, types of malicious URLs have become more diverse. Therefore, there is a need to consider the sustainability of a phishing detector by validating the detection models with consideration of the evolution of attacks. This can be done by selecting the best features that allow detectors to capture the dynamic nature of attacks. To validate the suitability, an ML model must be trained with samples captured during a specific time, and testing must be conducted for samples collected to represent future periods not involved in the training period.

VII. CONCLUSIONS

This article reviewed and analysed several research studies that combined the latest research in the field of detecting malicious URLs. The papers were recognized from a variety of articles obtained from reputable electronic sources. Primarily, this paper focused on reviewing studies about the detection of malicious URLs using ML algorithms, considering Arabic and non-Arabic content. The article presented several taxonomies and comparison results as a contribution to the field of malicious URLs detection. Additionally, the article highlighted and discussed several findings, including (1) lexical features of the URL, which is the most frequently used feature in both Arabic and non-Arabic content for detecting malicious URLs. Moreover, the studies conducted on Arabic websites did not utilize network-based features. (2) Regarding the detection techniques, the most frequently used algorithms in the reviewed papers were SVM, RF, and NB. Furthermore, the CNN and XGBoost models achieved higher performance than other algorithms, with an accuracy of 99.98%. (3) Regarding the datasets used, we found that most studies on Arabic content generated their own datasets, whereas the studies of non-Arabic content used open-source datasets like PhishTank for malicious web pages and Alexa for benign web pages. Finally, we discussed several challenges that might impact the quality of the ML detection techniques, including the size of the dataset, outliers, feature selection, and the sustainability of the detectors. This article can be considered a starting point for future research since it highlights the recent advancements and possible research directions.

ACKNOWLEDGMENT

The authors would like to thank SAUDI ARAMCO Cybersecurity Chair at Imam Abdulrahman Bin Faisal University for funding this project.

REFERENCES

[1] *Internet World Stats, "Top Ten Internet Languages in The World - Internet Statistics," Internet World Stats, 2020. https://www.internetworldstats.com/stats7.htm (accessed Oct. 14, 2021).*

[2] *M. E. Harsha Vardhan Sai Aalla, Nikhil Reddy Dumpala, "Malicious URL Prediction Using Machine Learning Techniques," Ann. Rom. Soc. Cell Biol., vol. 25, no. 5, pp. 2170–2176, 2021, Accessed: Jan. 19, 2022. [Online]. Available: https://www.annalsofrscb.ro/index.php/journal/article/view/4752 .*

[3] *"An Empirical analysis of phishing blacklists - Google Scholar." https://scholar.google.com/scholar?hl=en&as_sdt=0,5&q=An+Empirical+analysis+of+phishing+blacklists (accessed Apr. 04, 2022).*

[4] *H. Tupsamudre, A. K. Singh, and S. Lodha, "Everything is in the name – a URL based approach for phishing detection," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11527 LNCS, pp. 231–248, 2019, doi: 10.1007/978-3-030-20951-3_21.*

[5] *M. Aljabri and S. Mirza, "Phishing Attacks Detection using Machine Learning and Deep Learning Models," pp. 175–180, Mar. 2022, doi: 10.1109/cdma54072.2022.00034.*

[6] *M. Aljabri et al., "Intelligent Techniques for Detecting Network Attacks: Review and Research Directions," Sensors 2021, Vol. 21, Page 7070, vol. 21, no. 21, p. 7070, Oct. 2021, doi: 10.3390/S21217070.*

[7] *"Extracting Feature Vectors From URL Strings For Malicious URL Detection." https://towardsdatascience.com/extracting-feature-vectors-from-url-strings-for-malicious-url-detection-cbafc24737a (accessed Dec. 03, 2021).*

[8] *T. Manyumwa, P. F. Chapita, H. Wu, and S. Ji, "Towards Fighting Cybercrime: Malicious URL Attack Type Detection using Multiclass Classification," Proc. - 2020 IEEE Int. Conf. Big Data, Big Data 2020, pp. 1813–1822, Dec. 2020, doi: 10.1109/BIGDATA50022.2020.9378029.*

[9] *M. Alsaleh and A. Alarifi, "Analysis of web spam for non-English content: Toward more effective language-based classifiers," PLoS One, vol. 11, no. 11, Nov. 2016, doi: 10.1371/journal.pone.0164383.*

[10] *C. Johnson, B. Khadka, R. B. Basnet, and T. Doleck, "Towards detecting and classifying malicious urls using deep learning," J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl., vol. 11, no. 4, pp. 31–48, Dec. 2020, doi: 10.22667/JOWUA.2020.12.31.031.*

[11] *M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious JavaScript code," Proc. 19th Int. Conf. World Wide Web, WWW '10, pp. 281–290, 2010, doi: 10.1145/1772690.1772720.*

[12] *"Hacktivism and Website Defacement: Motivations, Capabilities and Potential Threats." https://www.researchgate.net/publication/320330579_Hacktivism_and_Website_Defacement_Motivations_Capabilities_and_Potential_Threats (accessed Nov. 25, 2021).*

[13] *M. Aldwairi and R. Alsalman, "MALURLS: A lightweight malicious website classification based on URL features," J. Emerg. Technol. Web Intell., vol. 4, no. 2, pp. 128–133, May 2012, doi: 10.4304/JETWI.4.2.128-133.*

[14] *"Alexa | Web Information Company's website." https://www.alexa.com/ (accessed Oct. 18, 2021).*

[15] *"PhishTank | Join the fight against phishing." https://www.phishtank.com/.*

[16] *C. Do Xuan, H. Dinh Nguyen, and T. Victor Nikolaevich, "Malicious URL Detection based on Machine Learning," IJACSA) Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 1, 2020.*

[17] *A. Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery, "Intelligent phishing website detection using random forest classifier," 2017 Int. Conf. Electr. Comput. Technol. Appl. ICECTA 2017, vol. 2018-Janua, pp. 1–5, Jun. 2017, doi: 10.1109/ICECTA.2017.8252051.*

[18] *M. Rami, M. Lee, and T. Fadi, "UCI Machine Learning Repository: Phishing Websites Data Set," 2015. https://archive.ics.uci.edu/ml/datasets/phishing+websites.*

[19] *J. Yuan, G. Chen, S. Tian, and X. Pei, "Malicious URL detection based on a parallel neural joint model," IEEE Access, vol. 9, pp. 9464–9472, 2021, doi: 10.1109/ACCESS.2021.3049625.*

[20] *Malware Domain List, "Malware Domain List," 2017. http://www.malwaredomainlist.com/mdl.php?inactive=on&sort=Reverse&search=&colsearch=All&ascordesc=ASC&quantity=50&page=318%0Ahttp://www.malwaredomainlist.com/mdl.php?search=125.19.103.198&colsearch=IP&quantity=200&inactive=on.*

[21] *"Alexa - Top sites." https://www.alexa.com/topsites (accessed Jan. 12, 2022).*

[22] *X. Yu, "Phishing Websites Detection Based on Hybrid Model of Deep Belief Network and Support Vector Machine," IOP Conference Series: Earth and Environmental Science, 2020.*

[23] *A. Zamir et al., "Phishing website detection using diverse machine learning algorithms," Electron. Libr., vol. 38, no. 1, pp. 65–80, Mar. 2020, doi: 10.1108/EL-05-2019-0118.*

[24] *K. Akash, "Phishing website dataset," Kaggle, 2018. https://www.kaggle.com/akashkr/phishing-website-dataset#dataset.csv/.*

[25] *F. Alkhudair, M. Alassaf, R. Ullah Khan, and S. Alfarraj, "Detecting Malicious URL," 2020 Int. Conf. Comput. Inf. Technol. ICCIT 2020, Sep. 2020, doi: 10.1109/ICCIT-144147971.2020.9213792.*

[26] *"Malicious and Benign Websites | Kaggle." https://www.kaggle.com/xwolf12/malicious-and-benign-websites (accessed Jan. 19, 2022).*

[27] *C. Camilo, U. López, J. O. Quintero, and A. Navarro, "Machine Learning Classifiers to Detect Malicious Websites," 2017, Accessed: Jan. 19, 2022. [Online]. Available: http://ceur-ws.org.*

[28] *G. Deebanchakkarawarthi, A. Parthan, L. Sachin, and A. Surya, "Classification of URL into Malicious or Benign using Machine Learning Approach," Int. J. Adv. Res. Comput. Commun. Eng., vol. 8, no. 2, 2019, doi: 10.17148/IJARCCE.2019.8247.*

[29] *S. G. Selvaganapathy, M. Nivaashini, and H. P. Natarajan, "Deep belief network based detection and categorization of malicious URLs," Inf. Secur. J., vol. 27, no. 3, pp. 145–161, May 2018, doi: 10.1080/19393555.2018.1456577.*

[30] *"UCI Machine Learning Repository: Spambase Data Set." https://archive.ics.uci.edu/ml/datasets/spambase (accessed Jan. 14, 2022).*

[31] *Netscape, "DMOZ OpenDirectoryProject." https://dmoz-odp.org/.*

[32] *R. S. Rao, A. R. Pais, and P. Anand, "A heuristic technique to detect phishing websites using TWSVM classifier," Neural Comput. Appl., vol. 33, no. 11, pp. 5733–5752, Jun. 2021, doi: 10.1007/s00521-020-05354-z.*

[33] *R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating deep learning approaches to characterize and classify malicious URL's," J. Intell. Fuzzy Syst., vol. 34, no. 3, pp. 1333–1343, 2018, doi: 10.3233/JIFS-169429.*

[34] *"Community Projects - RiskAnalytics." https://riskanalytics.com/community/ (accessed Jan. 19, 2022).*

[35] *"OpenPhish." https://openphish.com.*

[36] *D. R. Patil and J. B. Patil, "Feature-based Malicious URL and Attack Type Detection Using Multi-class Classification," vol. 10, no. 2, pp. 141–162, 2018, Accessed: Jan. 12, 2022. [Online]. Available: http://www.isecure-journal.org.*

[37] *"Spam domain blacklist (filtered by jwSpamSpy)." https://www.joewein.de/sw/blacklist.htm (accessed Jan. 12, 2022).*

[38] *P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," IEEE Access, vol. 7, pp. 15196–15209, 2019, doi: 10.1109/ACCESS.2019.2892066.*

[39] *"The directory of directories," Business Horizons, 1981. https://dmoztools.net/Computers/Software/Operating_Systems/Android/Markets/.*

[40] *Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, "Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network," Wirel. Commun. Mob. Comput., vol. 2021, 2021, doi: 10.1155/2021/8241104.*

[41] *Y. C. Chen, Y. W. Ma, and J. L. Chen, "Intelligent Malicious URL Detection with Feature Analysis," Proc. - IEEE Symp. Comput. Commun., vol. 2020-July, Jul. 2020, doi: 10.1109/ISCC50000.2020.9219637.*

[42] *"URL and website scanner - urlscan.io." https://urlscan.io/ (accessed Jan. 19, 2022).*

[43] *"Trending repositories on GitHub today · GitHub." https://github.com/trending.*

[44] *V. Vundavalli, F. Barsha, M. Masum, H. Shahriar, and H. Haddad, "Malicious URL Detection Using Supervised Machine Learning Techniques," Nov. 2020, doi: 10.1145/3433174.3433592.*

[45] *"Kaggle malicious dataset." https://www.kaggle.com/sid321axn/malicious-urls-dataset.*

[46] *A. Crisan, G. Florea, L. Halasz, C. Lemnaru, and C. Oprisa, "Detecting Malicious URLs Based on Machine Learning Algorithms and Word Embeddings," Proc. - 2020 IEEE 16th Int. Conf. Intell. Comput. Commun. Process. ICCP 2020, pp. 187–193, Sep. 2020, doi: 10.1109/ICCP51029.2020.9266139.*

[47] *J. Cao, Q. Li, Y. Ji, Y. He, and D. Guo, "Detection of Forwarding-Based Malicious URLs in Online Social Networks," Int. J. Parallel Program., vol. 44, no. 1, pp. 163–180, Feb. 2016, doi: 10.1007/s10766-014-0330-9.*

[48] *H. Faris and S. Yazid, "Phishing Web Page Detection Methods: URL and HTML Features Detection," IoTaIS 2020 - Proc. 2020 IEEE Int. Conf. Internet Things Intell. Syst., pp. 167–171, Jan. 2021, doi: 10.1109/IoTaIS50849.2021.9359694.*

[49] *R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," Neural Comput. Appl. 2018 318, vol. 31, no. 8, pp. 3851–3873, Jan. 2018, doi: 10.1007/S00521-017-3305-0.*

[50] *K. S. Adewole, A. G. Akintola, S. A. Salihu, N. Faruk, and R. G. Jimoh, "Hybrid Rule-Based Model for Phishing URLs Detection," Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST, vol. 285, pp. 119–135, Aug. 2019, doi: 10.1007/978-3-030-23943-5_9.*

[51] *Common Crawl, "Common Crawl," 2012. http://commoncrawl.org/.*

[52] *S. Kumi, C. Lim, and S.-G. Lee, "Malicious URL Detection Based on Associative Classification," Entropy, vol. 23, no. 2, p. 182, Jan. 2021, doi: 10.3390/e23020182.*

[53] *"VX Vault." http://vxvault.net/ViriList.php.*

[54] *abuse.ch, "URLhaus | Malware URL exchange," 2018. https://urlhaus.abuse.ch/.*

[55] *J. Liu, Y. Su, S. Lv, and C. Huang, "Detecting Web Spam Based on Novel Features from Web Page Source Code," Secur. Commun. Networks, vol. 2020, pp. 1–14, Dec. 2020, doi: 10.1155/2020/6662166.*

[56] *"GitHub - keras-team/keras: Deep Learning for humans." https://github.com/keras-team/keras (accessed Jan. 21, 2022).*

[57] *H. Wahsheh, I. A. Doush, M. Al-Kabi, I. Alsmadi, and E. Al-Shawakfa, "Using Machine Learning Algorithms to Detect Content-based Arabic Web Spam.," Journal of Information Assurance & Security, 2012. .*

[58] *A. S. Manjeri, R. Kaushik, A. Mnv, and P. C. Nair, "A Machine Learning Approach for Detecting Malicious Websites using URL Features," Proc. 3rd Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2019, pp. 555–561, Jun. 2019, doi: 10.1109/ICECA.2019.8821879.*

[59] *A. K. Singh, "Malicious and Benign Webpages Dataset," Data in Brief, 2020. https://ieee-dataport.org/documents/malicious-and-benign-websites.*

[60] *F. Vanhoenshoven, G. Napoles, R. Falcon, K. Vanhoof, and M. Koppen, "Detecting malicious URLs using machine learning techniques," 2016 IEEE Symp. Ser. Comput. Intell. SSCI 2016, Feb. 2017, doi: 10.1109/SSCI.2016.7850079.*

[61] *J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," ACM Int. Conf. Proceeding Ser., vol. 382, 2009, doi: 10.1145/1553374.1553462.*

[62] *A. C. Rakotoasimbahoaka, I. Randria, and N. R. Razafindrakoto, "Malicious URL detection Using majority vote method with machine learning and deep learning models," Proc. - 2020 Int. Conf. Interdiscip. Cyber Phys. Syst. ICPS 2020, pp. 37–43, Dec. 2020, doi: 10.1109/ICPS51508.2020.00013.*

[63] *P. V. Rao, S. G. Rao, P. C. Reddy, B. S. A. Kumar, and G. A. Kumar, "Detection of Malicious uniform Resource Locator," Int. J. Recent Technol. Eng., no. 2, pp. 2277–3878, 2019, doi: 10.35940/ijrte.A1265.078219.*

[64] *A. Rakotoasimbahoaka, I. Randria, and R. Razafindrakoto, "Malicious URL Detection by Combining Machine Learning and Deep Learning Models," Malicious URL Detect. by Comb. Mach. Learn. Deep Learn. Model.*

[65] *R. Chiramdasu, G. Srivastava, S. Bhattacharya, P. K. Reddy,*

and T. Reddy Gadekallu, "Malicious url detection using logistic regression," 2021 IEEE Int. Conf. Omni-Layer Intell. Syst. COINS 2021, Aug. 2021, doi: 10.1109/COINS51742.2021.9524269.

[66] *Y. Shi, G. Chen, and J. Li, "Malicious Domain Name Detection Based on Extreme Machine Learning," Neural Process. Lett., vol. 48, no. 3, pp. 1347–1357, Dec. 2018, doi: 10.1007/s11063-017-9666-7.*

[67] *S. Parekh, D. Parikh, S. Kotak, and S. Sankhe, "A New Method for Detection of Phishing Websites: URL Detection," Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2018, pp. 949–952, Sep. 2018, doi: 10.1109/ICICCT.2018.8473085.*

[68] *A. Butnaru, A. Mylonas, and N. Pitropakis, "Towards lightweight url-based phishing detection," Futur. Internet, vol. 13, no. 6, p. 154, Jun. 2021, doi: 10.3390/fi13060154.*

[69] *S. Kumar, "Malicious And Benign URLs | Kaggle." https://www.kaggle.com/siddharthkumar25/malicious-and-benign-urls.*

[70] *Shantanu, B. Janet, and R. Joshua Arul Kumar, "Malicious URL Detection: A Comparative Study," Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021, pp. 1147–1151, Mar. 2021, doi: 10.1109/ICAIS50930.2021.9396014.*

[71] *A. Astorino, A. Chiarello, M. Gaudioso, and A. Piccolo, "Malicious URL detection via spherical classification," Neural Comput. Appl., vol. 28, pp. 699–705, Dec. 2017, doi: 10.1007/s00521-016-2374-9.*

[72] *"OpenDirectory." https://opendirectory.org/ (accessed Jan. 30, 2022).*

[73] *Y. Peng, S. Tian, L. Yu, Y. Lv, and R. Wang, "A Joint Approach to Detect Malicious URL Based on Attention Mechanism," Int. J. Comput. Intell. Appl., vol. 18, no. 3, Sep. 2019, doi: 10.1142/S1469026819500214.*

[74] *D. J. Wadas, "Detecting Phishing URLs Using Machine Learning Techniques," 2019.*

[75] *"Ebubekirbbr (n.d.). pdd. Retrieved from GitHub." https://github.com/ebubekirbbr/pdd/tree/master/input (accessed Jan. 17, 2022).*

[76] *F. Sadique, R. Kaul, S. Badsha, and S. Sengupta, "An Automated Framework for Real-time Phishing URL Detection," 2020 10th Annu. Comput. Commun. Work. Conf. CCWC 2020, pp. 335–341, Jan. 2020, doi: 10.1109/CCWC47524.2020.9031269.*

[77] *R. Patgiri, H. Katari, R. Kumar, and D. Sharma, "Empirical study on malicious URL detection using machine learning," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11319 LNCS, pp. 380–388, 2019, doi: 10.1007/978-3-030-05366-6_31.*

[78] *D. C. C. Rupa, G. Srivastava, S. Bhattacharya, P. Reddy, and T. R. R. Gadekallu, "A Machine Learning Driven Threat Intelligence System for Malicious URL Detection," ACM Int. Conf. Proceeding Ser., vol. 7, pp. 1–7, Aug. 2021, doi: 10.1145/3465481.3470029.*

[79] *J. C. Prieto, A. Fernandez-Isabel, I. M. De Diego, F. Ortega, and J. M. Moguerza, "Knowledge-Based Approach to Detect Potentially Risky Websites," IEEE Access, vol. 9, pp. 11633–11643, 2021, doi: 10.1109/ACCESS.2021.3051374.*

[80] *A. Desai, J. Jatakia, R. Naik, and N. Raul, "Malicious web content detection using machine learning," RTEICT 2017 - 2nd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. Proc., vol. 2018-Janua, pp. 1432–1436, Jul. 2017, doi: 10.1109/RTEICT.2017.8256834.*

[81] *"UCI Machine Learning Repository: Data Set." https://archive.ics.uci.edu/ml/datasets/phishing+websites.*

[82] *I. Akour, A. Aburayya, D. Health Authority, and R. Alfaisal, "USING CLASSICAL MACHINE LEARNING FOR PHISHING WEBSITES DETECTION FROM URLS," 2021.*

[83] *H. Yuan, Z. Yang, X. Chen, Y. Li, and W. Liu, "URL2Vec: URL modeling with character embeddings for fast and accurate phishing website detection," Proc. - 16th IEEE Int. Symp. Parallel Distrib. Process. with Appl. 17th IEEE Int. Conf. Ubiquitous Comput. Commun. 8th IEEE Int. Conf. Big Data Cloud Comput. 11t, pp. 265–272, 2019, doi: 10.1109/BDCloud.2018.00050.*

[84] *S. He, J. Xin, H. Peng, and E. Zhang, "Research on Malicious URL Detection Based on Feature Contribution Tendency," in 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics, ICCCBDA 2021, Apr. 2021, pp. 576–581, doi: 10.1109/ICCCBDA51879.2021.9442606.*

[85] *"Malwaredomainlist." http://www.malwaredomainlist.com/.*

[86] *"Cybercrime-tracker." http://cybercrime-tracker.net/.*

[87] *"360.com." https://data.netlab.360.com/.*

[88] *A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN–LSTM model for detecting phishing URLs," Neural Comput. Appl., pp. 1–17, Aug. 2021, doi: 10.1007/S00521-021-06401-Z/TABLES/7.*

[89] *"pdd/input at master · ebubekirbbr/pdd · GitHub." https://github.com/ebubekirbbr/pdd/tree/master/input (accessed Jan. 15, 2022).*

[90] *S. Marchal, J. Francois, R. State, and T. Engel, "Phish storm: Detecting phishing with streaming analytics," IEEE Trans. Netw. Serv. Manag., vol. 11, no. 4, pp. 458–471, Dec. 2014, doi: 10.1109/TNSM.2014.2377295.*

[91] *O. V. Lee et al., "A malicious URLs detection system using optimization and machine learning classifiers," Indones. J. Electr. Eng. Comput. Sci., vol. 17, no. 3, pp. 1210–1214, Mar. 2019, doi: 10.11591/ijeecs.v17.i3.pp1210-1214.*

[92] *A. Saleem Raja, R. Vinodini, and A. Kavitha, "Lexical features based malicious URL detection using machine learning techniques," Mater. Today Proc., Apr. 2021, doi: 10.1016/j.matpr.2021.04.041.*

[93] *"URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB." https://www.unb.ca/cic/datasets/url-2016.html.*

[94] *V. N and V. V, "Malicious-URL Detection using Logistic Regression Technique," Int. J. Eng. Manag. Res., vol. 09, no. 06, pp. 108–113, Dec. 2019, doi: 10.31033/ijemr.9.6.18.*

[95] *"yahoo-phishing | GitHub." https://github.com/topics/yahoo-phishing.*

[96] *J. H. Ateeq and M. Moreb, "Detecting Malicious URL using Neural Network," in 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Jul. 2021, pp. 1–8, doi: 10.1109/ICOTEN52080.2021.9493481.*

[97] *"Investigation on Android Malware 2019 | Datasets | Research | Canadian Institute for Cybersecurity | UNB." https://www.unb.ca/cic/datasets/invesandmal2019.html.*

[98] *S. Shivangi, P. Debnath, K. Saieevan, and D. Annapurna, "Chrome Extension for Malicious URLs detection in Social Media Applications Using Artificial Neural Networks and Long Short Term Memory Networks," in 2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018, Nov. 2018, pp. 1993–1997, doi: 10.1109/ICACCI.2018.8554647.*

[99] *Y. Pingle, S. Patil, S. N. Bhatkar, and S. Patil, "Detection of Maliciious Content using AI," 7th Int. Conf. "Computing Sustain. Glob. Dev., 2020, [Online]. Available: http://bvicam.in/INDIACom/news/INDIACom 2020 Proceedings/Main/papers/33.pdf.*

[100] *A. Lakshmanarao, M. R. Babu, and M. M. Bala Krishna, "Malicious URL Detection using NLP, Machine Learning and FLASK," 2021 Int. Conf. Innov. Comput. Intell. Commun. Smart Electr. Syst., pp. 1–4, Sep. 2021, doi: 10.1109/ICSES52305.2021.9633889.*

[101] *"Phishing Site URLs | Kaggle." https://www.kaggle.com/taruntiwarihp/phishing-site-urls (accessed Jan. 29, 2022).*

[102] *H. M. J. Khan, Q. Niyaz, V. K. Devabhaktuni, S. Guo, and U. Shaikh, "Identifying Generic Features for Malicious URL Detection System," 2019 IEEE 10th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2019, pp. 0347–0352, Oct. 2019, doi: 10.1109/UEMCON47517.2019.8992930.*

[103] *"Malicious_n_Non-Malicious URL | Kaggle." https://www.kaggle.com/antonyj453/urldataset (accessed Jan. 29, 2022).*

[104] *M. Abutaha, M. Ababneh, K. Mahmoud, and S. A. H. Baddar, "URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis," in 2021 12th International Conference on Information and Communication Systems, ICICS 2021, May 2021, pp. 147–152, doi: 10.1109/ICICS52457.2021.9464539.*

[105] *RLILOJR, "rlilojr/Detecting-Malicious-URL-Machine-Learning." https://github.com/rlilojr/Detecting-MaliciousURL-Machine-Learning (accessed Nov. 14, 2020).*

[106] *J. Zhao, N. Wang, Q. Ma, and Z. Cheng, "Classifying malicious urls using gated recurrent neural networks," in Advances in Intelligent Systems and Computing, 2019, vol. 773, pp. 385–394, doi: 10.1007/978-3-319-93554-6_36.*

[107] *B. Cui, S. He, P. Shi, and X. Yao, "Malicious URL detection with feature extraction based on machine learning," Int. J. High Perform. Comput. Netw., vol. 12, no. 2, pp. 166–178, 2018, doi: 10.1504/ijhpcn.2018.094367.*

[108] *J. Yang, P. Yang, X. Jin, and Q. Ma, "Multi-Classification for Malicious URL Based on Improved Semi-Supervised Algorithm," in Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017, Aug. 2017, vol. 1, pp. 143–150, doi: 10.1109/CSE-EUC.2017.34.*

[109] *Q. T. Hai and S. O. Hwang, "Detection of malicious URLs based on word vector representation and ngram," J. Intell. Fuzzy Syst., vol. 35, no. 6, pp. 5889–5900, 2018, doi: 10.3233/JIFS-169831.*

[110] *"Malc0de Database." http://malc0de.com/database/ (accessed Jan. 12, 2022).*

[111] *"CleanMX url database." http://support.clean-mx.com.*

[112] *B. Banik and A. Sarma, "Phishing URL detection system based on URL features using SVM," Int. J. Electron. Appl. Res., vol. 5, no. 2, pp. 40–55, Dec. 2018, doi: 10.33665/ijear.2018.v05i02.003.*

[113] *"DMOZ directory." https://github.com/gr33ndata/dmoz-urlclassifier/.*

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

[115] *A. Correa Bahnsen, "DeepPhish Simulating Malicious AI," Proc. Symp. Electron. Crime Res. San Diego, CA, USA, pp. 15–17, 2018, [Online]. Available: https://i.blackhat.com/eu-18/Wed-Dec-5/eu-18-CorreaBahnsen-DeepPhish-Simulating-Malicious-AI.pdf.*

[116] *Y. Liang and X. Yan, "Using deep learning to detect malicious URLs," in Proceedings - IEEE International Conference on Energy Internet, ICEI 2019, May 2019, pp. 487–492, doi: 10.1109/ICEI.2019.00092.*

[117] *360NetLab, "DGA | Netlab OpenData Project," Qihoo 360 Technology, 2016. http://data.netlab.360.com/dga/.*

[118] *Alexa Internet Inc., "Alexa top 1 million sites," Kaggle Datasets, 2019. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.*

[119] *A. Joshi, L. Lloyd, P. Westin, and S. Seethapathy, "Using Lexical Features for Malicious URL Detection -- A Machine Learning Approach," Oct. 2019.*

[120] *J. Ispahany and R. Islam, "Detecting malicious COVID-19 URLs using machine learning techniques," in 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, PerCom Workshops 2021, Mar. 2021, pp. 718–723, doi: 10.1109/PerComWorkshops51409.2021.9431064.*

[121] *J. Abrams, "Free COVID-19 Threat List - Domain Risk Assessments for Coronavirus Threats," Jun. 2020. https://www.domaintools.com/resources/blog/free-covid-19-threat-list-domain-risk-assessments-for-coronavirus-threats.*

[122] *"Whois Domain Lookup." https://www.whois.com/whois.*

[123] *S. Afzal, M. Asim, A. R. Javed, M. O. Beg, and T. Baker, "URLdeepDetect: A Deep Learning Approach for Detecting*

Malicious URLs Using Semantic Vector Models," J. Netw. Syst. Manag., vol. 29, no. 3, Jul. 2021, doi: 10.1007/S10922-021-09587-8.

[124] *Y. Zeng, "Malicious URLs and Attachments Detection on Lexical-based Features using Machine Learning Techniques," 2018.*

[125] *B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," Comput. Commun., vol. 175, pp. 47–57, 2021, doi: 10.1016/j.comcom.2021.04.023.*

[126] *B. Banik and A. Sarma, "Lexical Feature Based Feature Selection and Phishing URL Classification Using Machine Learning Techniques," Commun. Comput. Inf. Sci., vol. 1241 CCIS, pp. 93–105, Jul. 2020, doi: 10.1007/978-981-15-6318-8_9.*

[127] *K. L. Chiew et al., "Building Standard Offline Anti-phishing Dataset for Benchmarking," Int. J. Eng. Technol., vol. 7, no. 4.31, pp. 7–14, Dec. 2018, doi: 10.14419/ijet.v7i4.31.23333.*

[128] *B. Banik and A. Sarma, "Phishing URL detection system based on URL features using SVM," International Journal of Electronics and Applied Research, 2018.*

[129] *O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," Expert Syst. Appl., vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/J.ESWA.2018.09.029.*

[130] *"Yandex.XML — Yandex Teknolojileri." https://yandex.com.tr/dev/xml/.*

[131] *A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. Gonzalez, "Classifying phishing URLs using recurrent neural networks," in eCrime Researchers Summit, eCrime, Jun. 2017, pp. 1–8, doi: 10.1109/ECRIME.2017.7945048.*

[132] *W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," Comput. Networks, vol. 178, no. January, 2020, doi: 10.1016/j.comnet.2020.107275.*

[133] *"Technical challenge of network security." https://www.kesci.com/apps/home/dataset/58f32a96a686fb29e425a567.*

[134] *"Reasonable Antiphishing," [Online]. Available: http://antiphishing.reasonables.com/BlackList.aspx.*

[135] *R. Yang, K. Zheng, B. Wu, C. Wu, and X. Wang, "Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning," Sensors, vol. 21, no. 24, p. 8281, Dec. 2021, doi: 10.3390/S21248281.*

[136] *"Yandex.Toloka Open Datasets." https://research.yandex.com/datasets/toloka (accessed Jan. 16, 2022).*

[137] *J. Yuan, Y. Liu, and L. Yu, "A Novel Approach for Malicious URL Detection Based on the Joint Model," Secur. Commun. Networks, vol. 2021, pp. 1–12, Dec. 2021, doi: 10.1155/2021/4917016.*

[138] *"Hphosts." https://www.hosts-file.net/.*

[139] *B. Altay, T. Dokeroglu, and A. Cosar, "Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection," Soft Comput., vol. 23, no. 12, pp. 4177–4191, Jun. 2019, doi: 10.1007/s00500-018-3066-4.*

[140] *"Cybersecurity to Prevent Breaches.| Comodo Cybersecurity." https://www.comodo.com.*

[141] *J. McGahagan, D. Bhansali, C. Pinto-Coelho, and M. Cukier, "A Comprehensive Evaluation of Webpage Content Features for Detecting Malicious Websites," Nov. 2019, doi: 10.1109/LADC48089.2019.8995713.*

[142] *talosintelligence.com, "Snort || Cisco Talos Intelligence Group - Comprehensive Threat Intelligence," Cisco, 2020. https://talosintelligence.com/snort.*

[143] *A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," J. Ambient Intell. Humaniz. Comput., vol. 10, no. 5, pp. 2015–2028, May 2019, doi: 10.1007/S12652-018-0798-Z.*

[144] *"Welcome to CentOS." http://www.stuffgate.com/ (accessed Jan. 19, 2022).*

[145] *M. Al-Kabi, H. Wahsheh, I. Alsmadi, E. Al-Shawakfa, A. Wahbeh, and A. Al-Hmoud, "Content-based analysis to detect Arabic web spam," J. Inf. Sci., vol. 38, no. 3, pp. 284–296, Jun. 2012, doi: 10.1177/0165551512439173.*

[146] *I. Alsmadi, "The automatic evaluation of website metrics and state," Int. J. Web-Based Learn. Teach. Technol., vol. 5, no. 4, pp. 1–17, 2010, doi: 10.4018/jwltt.2010100101.*

[147] *M. N. Al-Kabi, H. A. Wahsheh, and I. M. Alsmadi, "OLAWSDS: An Online Arabic Web Spam Detection System," 2014.*

[148] *E. M., A. F., and H. E., "Web Mining Techniques to Block Spam Web Sites," Int. J. Comput. Appl., vol. 181, no. 8, pp. 36–42, Aug. 2018, doi: 10.5120/ijca2018917622.*

[149] *"قائمة المواقع المحجوبة في مصر - مؤسسة حرية الفكر والتعبير" https://afteegypt.org/blocked-websites-list-ar (accessed Jan. 19, 2022).*

[150] *H. A. Wahsheh, M. N. Al-Kabi, and I. M. Alsmadi, "A link and Content Hybrid Approach for Arabic Web Spam Detection," Int. J. Intell. Syst. Appl., vol. 5, no. 1, pp. 30–43, Dec. 2012, doi: 10.5815/ijisa.2013.01.03.*

[151] *N. Al-Twairesh, M. Al-Tuwaijri, A. Al-Moammar, and S. Al-Humoud, "Arabic Spam Detection in Twitter," 2nd Work. Arab. Corpora Process. Tools 2016 Theme Soc. Media , pp. 38–43, May 2016.*

[152] *D. Alorini, "Towards Machine Learning for Gulf Dialectical Arabic Malicious Content Detection in Social Media," 2018.*

[153] *M. Alkhair, K. Meftouh, K. Smaïli, and N. Othman, "An Arabic Corpus of Fake News: Collection, Analysis and Classification," in Communications in Computer and Information Science, 2019, vol. 1108, pp. 292–302, doi: 10.1007/978-3-030-32959-4_21.*

[154] *H. A. Wahsheh, M. N. Al-kabi, and I. M. Alsmadi, "Evaluating Arabic spam classifiers using link analysis," 2012, doi:*

10.1145/2222444.2222456.

[155] "Link Checker, Web Link Validator." http://www.relsoftware.com/ (accessed Apr. 11, 2022).

[156] M. Mataoui, O. Zelmati, D. Boughaci, M. Chaouche, and F. Lagoug, "A proposed spam detection approach for Arabic social networks content," in Proceedings of the 2017 International Conference on Mathematics and Information Technology, ICMIT 2017, Jul. 2017, vol. 2018-Janua, pp. 222–226, doi: 10.1109/MATHIT.2017.8259721.

[157] A. R. Alharbi and A. Aljaedi, "Predicting rogue content and arabic spammers on twitter," Futur. Internet, vol. 11, no. 11, p. 21, Nov. 2019, doi: 10.3390/FI11110229.

[158] H. Najadat, M. A. Alzubaidi, and I. Qarqaz, "Detecting Arabic Spam Reviews in Social Networks Based on Classification Algorithms," ACM Trans. Asian Low-Resource Lang. Inf. Process., vol. 21, no. 1, pp. 1–13, Jan. 2022, doi: 10.1145/3476115.

[159] "up2 | Netvizz." http://www.up2.fr/index.php?n=Main.Netvizz.

[160] H. Mubarak, A. Abdelali, S. Hassan, and K. Darwish, "Spam Detection on Arabic Twitter," 2020, pp. 237–251.

[161] M. M. Alsulami and A. Yousef, "SentiFilter: A Personalized Filtering Model for Arabic Semi-Spam Content based on Sentimental and Behavioral Analysis," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 2, 2020, doi: 10.14569/IJACSA.2020.0110218.

[162] H. A. Wahsheh, M. N. Al-Kabi, and I. M. Alsmadi, "SPAR: A system to detect spam in Arabic opinions," 2013, doi: 10.1109/AEECT.2013.6716442.

[163] M. A. AlGhamdi and M. A. Khan, "Intelligent Analysis of Arabic Tweets for Detection of Suspicious Messages," Arab. J. Sci. Eng., vol. 45, no. 8, pp. 6021–6032, Aug. 2020, doi: 10.1007/s13369-020-04447-0.

[164] E. S. Grange et al., "Responding to COVID-19: The UW Medicine Information Technology Services Experience," Appl. Clin. Inform., vol. 11, no. 2, pp. 265–275, Mar. 2020, doi: 10.1055/S-0040-1709715.

[165] "XGBoost: Everything You Need to Know - neptune.ai." https://neptune.ai/blog/xgboost-everything-you-need-to-know (accessed Jul. 10, 2022).

[166] "Difference Between Machine Learning and Deep Learning - GeeksforGeeks." https://www.geeksforgeeks.org/difference-between-machine-learning-and-deep-learning/ (accessed Jul. 10, 2022).

[167] "Predictive Model Ensembles: Pros and Cons - Perficient Blogs." https://blogs.perficient.com/2019/11/07/predictive-model-ensembles-pros-and-cons/ (accessed Jul. 10, 2022).

[168] "Neural Network Applications in E-Commerce: Advantages & Disadvantages." https://blog.clerk.io/neural-network (accessed Jul. 10, 2022).

[169] "URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB." https://www.unb.ca/cic/datasets/index.html.

[170] "Company History | OpenDNS." https://www.opendns.com/about/company-history/ (accessed Jul. 08, 2022).

[171] "Cisco Completes Acquisition of OpenDNS." https://newsroom.cisco.com/c/r/newsroom/en/us/a/y2015/m08/cisco-completes-acquisition-of-opendns.html (accessed Jul. 08, 2022).

[172] "ALEXA Internet Donates Archive of the World Wide Web To Library of Congress." https://web.archive.org/web/20091013152257/http://www.loc.gov/today/pr/1998/98-167.html (accessed Jul. 08, 2022).

[173] "Alexa.com acquired by Amazon." https://www.crunchbase.com/acquisition/amazon-acquires-alexa--dad2c8ff (accessed Jul. 08, 2022).

[174] "What is Alexa Rank and Its Value?" https://attentioninsight.com/what-is-alexa-rank-and-its-value/ (accessed Jul. 08, 2022).

Biographies

Dr. Malak Aljabri received the B.S degree in computer science from Umm Al-Qura University, Saudi Arabia in 2006, and the M.S degree in Computer Science from Heriot watt University, UK, in 2010. The Ph.D. also in Computer Science from the University of Glasgow, United Kingdom, in 2015. She is an Assistant Professor at the College of Computer Science and Information Technology CCSIT, at Imam Abdulrahman bin Faisal University, Saudi Arabia.

Dr. Amal Alahmadi received the B.S and M.S degree in computer science from King Abdulaziz University, Saudi Arabia in 2001 and 2008, respectively. She received the Ph.D. in Communications and Networks Engineering from the University of Leeds, United Kingdom, in 2020. She is an Assistant Professor at the College of Computer Science and Information Technology CCSIT, at Imam Abdulrahman bin Faisal University, Saudi Arabia.

Dr. Fahd Alhaidari received the B.S degree in computer science from Mousl University, Iraq, in 1999. He received his M.S degree and Ph.D. degree in Computer Science from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in 2007, and 2011, respectively. He is an Assistant Professor at the College of Computer Science and Information Technology CCSIT, at Imam Abdulrahman bin Faisal University, Saudi Arabia.

Dr. Rami Mustafa received the B.S degree in Computer Science and Information Systems from Philadelphia University, Jordan, in 1998, and the M.S in Information Systems from Araba Academy for Banking and Financial Sciences, Jordan, in 2002, and the Ph.D. in Computer Science and Informatics from the University of Huddersfield, United Kingdom, in 2016. He is an Assistant Professor at the College of Computer Science and Information Technology CCSIT, at Imam Abdulrahman bin Faisal University, Dammam, Eastern Province, Saudi Arabia.

Dr. Khaled Salah received the B.S. degree in computer engineering with a minor in computer science from Iowa State University, USA, in 1990, and the M.S. degree in computer systems engineering and the Ph.D. degree in computer science from the Illinois Institute of Technology, USA, in 1994 and 2000, respectively. He is a Full Professor at the Department of Electrical and Computer Engineering, Khalifa University, United Arab Emirates. He has over 220 publications and three U.S. patents, has been giving several international keynote speeches, invited talks, tutorials, and research seminars on blockchain, the IoT, fog and cloud computing, and cybersecurity. He is now leading several projects on how to leverage blockchain for healthcare, 5G networks, combating deep fake videos, supply chain management, and AI. He served as the Chair of the Track Chairs for IEEE Globecom 2018 on Cloud Computing. He is an Associate Editor of IEEE Blockchain Tech Briefs and a member of IEEE Blockchain Education Committee. (Based on a document published on 22 December 2021)

Hanan Altamimi received the B.S degree in computer science from Imam Abdulrahman bin Faisal University, Saudi Arabia, in 2022.

Maimunah AL-Harbi received the B.S degree in computer science from Imam Abdulrahman bin Faisal University, Saudi Arabia, in 2022.

Najd Alotaibi received the B.S degree in computer science from Imam Abdulrahman bin Faisal University, Saudi Arabia, in 2022.

Haya Alhuraib received the B.S degree in computer science from Imam Abdulrahman bin Faisal University, Saudi Arabia, in 2022.

Shahd Albelali received the B.S degree in computer science from Imam Abdulrahman bin Faisal University, Saudi Arabia, in 2022.