IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 July 2025, accepted 14 August 2025, date of publication 21 August 2025, date of current version 8 September 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3601387

SURVEY

"Think Before You Click"–Malicious URL Detection in Cybersecurity: A Systematic Review and Research Roadmap

SAUREABH KAILAS AND R. ROOPALAKSHMI
Department of CSE, Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Manipal, Karnataka 576104, India
Corresponding author: R. Roopalakshmi (roopalakshmi.r@manipal.edu)

ABSTRACT **Introduction:** Thanks to the world wide web (www) revolutions and COVID-19 pandemic, the dependency on digital platforms has tremendously increased and making them essential for day-to-day information access. Although, technological advancements boosted digital platform usage, but also increased user's vulnerability to cyberattacks. **Background:** For instance, In India, the reported cyber crime complaints increased by 15.3% in Quarter-2 of 2022, totaling 237,658 complaints. Further, Phishing attacks, are responsible for over 90% of data breaches, which are the most common and severe cyber threats. Despite the diversity of cyber-threats, URLs serve as the primary gateway in most attacks, making their detection crucial for protecting users. **Methodology:** This article provides a comprehensive literature review on malicious URL detection techniques, including Listing, Heuristics, Machine Learning, Feature Engineering and Emerging Methods, by highlighting limitations, feature types, and datasets and thereby presents the broader perspective compared to contemporary single-methodology-based review studies. Further, this article follows the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (**PRISMA**) model for a thorough, transparent review process. **Results:** Based on the research study, this article highlights unexplored research challenges associated with each category of detection techniques, feature types, which play a crucial role in effective malicious URL detection. It also emphasizes on Theoretical, Managerial implications of this study, Real-world deployment constraints, Research Roadmap and thereby encourages the future researchers to address these challenges and develop innovative solutions.

INDEX TERMS Malicious URL detection, PRISMA model, machine learning, blacklisting techniques, whitelisting techniques, deep learning, feature engineering, URL features.

I. INTRODUCTION

In this Internet era, thanks to world wide web (www) revolutions, which is emerging as the mandatory ones for almost all kinds of information access, communication and transaction services. In addition, technological advancements and the COVID-19 pandemic situations hugely accelerated the dependency on digital platforms for day-to-day activities, which in turn made the users more susceptible to cyber threats. For instance, as per the latest National Cybercrime Reporting Portal (NCRP) of Indian Govt. there is a rise of 15.3% in reported complaints during II-Quarter of 2022, which accounts for a total of 2,37,658 complaints, when compared to I-Quarter of 2022 [1]. Specifically, as per the NCRP report, '*Online Financial Fraud*', is the most prevalent cyber crime category, which accounts for almost 67.9% of the total reported cyber crimes in India [1].

In general, common cyber threats include phishing attacks, spam mails and malware distribution through webpages, which eventually result in significant financial losses [2]. For instance, phishing results in financial fraud by exploiting stolen credentials and payment information, whereas spam emails results in promotion of unauthorized products and stealing of personal information. Malware distributions

The associate editor coordinating the review of this manuscript and approving it for publication was Hai Dong.

include activities such as installing ransomware or virus on victim computers for system hijacking and identity theft purposes [2]. Among all the cyber threats, Phishing attacks are the most common and serious ones, which constitutes over 90% of the data breaches [3]. For example, as per the latest 2022-report, majority of online security breaches involve phishing attacks, which contribute to almost 78% of digital surveillance assaults [3]. Further, according to 2021-survey, cyber crime losses are projected to reach up to 6 trillion dollars annually world-wide, contributing to a global cost of 11.4 million dollars [4].

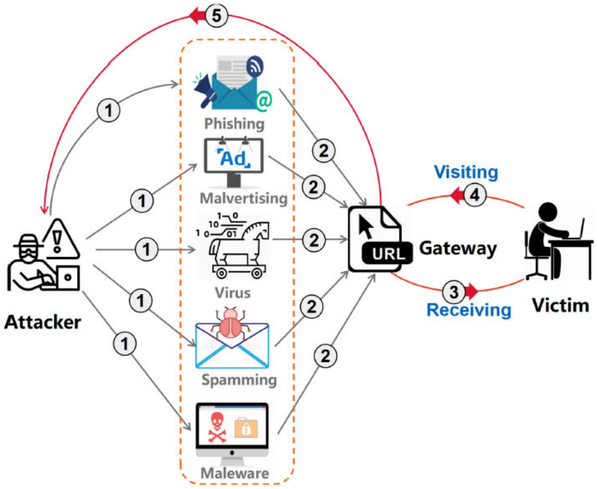


FIGURE 1. Malicious URLs act as primary gateways to diverse cyber attacks [4].

Although cyber attacks are diverse in nature; yet, the Uniform Resource Locators (URLs) serve as the primary gateways or medium, using which the malicious agents link with the victim users. Specifically, as shown in Figure 1. malicious URLs act as the main gateway for the attackers in order to link with compromised websites and forged accounts. More specifically, most of the potential users struggle to distinguish between such malicious URLs from the legitimate ones. Therefore, detecting malicious URLs in advance, is very much crucial to protect the vulnerable end users from various types of cyber threats. For instance, in case of search engine spamming or URL obfuscation attacks, by the time the attack is confirmed, it might have resulted in extensive unsolicited exposure of user credentials [5], [6]. Due to these reasons, promising techniques, which can automatically detect malicious URLs, before the user accesses them are very much essential, so that they can significantly strengthen the cyber defense and thereby prevent extensive exposure of victims credentials [7]. Based on these aspects, this research article, presents an extensive review of state-of-the-art malicious URL detection techniques in terms of different categories including Listing techniques, Heuristics approaches, Machine learning, Deep learning and Feature-engineering techniques along with Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [8] flow diagrams. Specifically, this review article presents the taxonomy of existing malicious URL identification approaches along with their methodology descriptions and experimental datasets. This research article also highlights the key research directions in each category of the malicious URL detection techniques, which are yet to be explored and remain as open research challenges in this research domain.

Due to the broad nature of this malicious URL detection problem, this review article is structured as follows: Section II illustrates the different methodologies employed in the malicious URL detection domain by means of presenting the review of literature, Taxonomy of techniques and PRISMA flow diagrams. Section III describes some of the important features used in the malicious URL prediction domain including URL-based, content-based, lexical and other features. Section IV presents the publicly available datasets, which are widely popular in this domain. Section V highlights the primary research challenges in each of the methodologies, which are yet to be explored by the future research communities along with the emerging methods in Section VI, followed by theoretical, managerial implications, deployment challenges, research roadmap in Section VII and conclusion and future work in Section VIII.

A. MOTIVATION AND CONTRIBUTIONS

Considering the complex and extensive nature of the malicious URL detection problem, the **Key Contributions of this review article** are highlighted as follows:

1) **Introducing Structured Taxonomy of Techniques:** After conducting a comprehensive review on the state-of-the-art literature of malicious URLs detection techniques, this article presents a structured taxonomy of existing techniques.
2) **In-depth Literature Survey:** This article presents an in-depth literature survey on malicious webpage identification methods, which comprises techniques such as Whitelisting, Blacklisting, Heuristic approaches, Machine learning, feature engineering, Emerging Methods along with their respective subcategories and relevant existing research frameworks.
3) **Presenting PRISMA Flow diagram:** Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) model [8] is employed in this article for ensuring a thorough as well as transparent presentation of the systematic review process.
4) **Methodology-wise Distribution of articles:** This research study presents a tabulation of articles, which indicates a clear-cut breakdown of the articles by methodology-wise along with their respective subcategories, datasets, features, performance metrics and associated research studies, so that the clarity and organization of this study can be enhanced.
5) **Summary of Quantitative Meta-Analysis of Performance Metrics:** This study presents the Summary of

Quantitative Meta-Analysis of Performance Metrics of existing studies, which includes details of representative methods, datasets, performance metrics and their trade-offs. Specifically, performance metrics such as Accuracy, F1-score, ROC-AUC of representative methods are highlighted in a summarized way, which will benefit the future studies in developing innovative solutions.

6) **Exhaustive Review compared to its Contemporaries:** This article represents an exhaustive literature review that systematically classifies all the malicious URL detection frameworks whereas the existing review-studies mainly focus only on specific category of techniques. Precisely, most of the contemporary reviews are primarily focusing only on Machine Learning or Deep Learning approaches [3], [7], [10], [3], whereas this research review attempts to explore all the existing category of techniques along with their feature types and datasets.
7) **More Focus on Categorization of URL Features:** This article organizes and categorizes the essential features used in malicious URL detection problem, into distinct groups, such as content-based, URL-based, lexical, domain-specific and other relevant features, along with the corresponding research studies, which is underexplored in this domain.
8) **Extensive Dataset Descriptions:** The article identifies and illustrates most popular publicly-available datasets in the malicious URL detection domain and thereby serves as an essential resource for future research studies.
9) **Outlining Unexplored Research Challenges and Implications:** This article highlights unexplored research challenges associated with each category of detection techniques, feature types which play a crucial role in effective malicious URL detection. It also emphasizes on Theoretical, Managerial implications of this study, Real-world deployment constraints, Research Roadmap and thereby encourages the future researchers to address these challenges and develop innovative solutions.

II. MALICIOUS URLs DETECTION: LITERATURE REVIEW

From the past few decades, a lot of research studies are carried out towards the identification of malicious web pages, which are illustrated in the forthcoming subsections.

A. TAXONOMY OF MALICIOUS URL DETECTION TECHNIQUES

The state-of-the-art research studies on malicious URL detection, in the broader sense, can be categorized into four major groups namely, URL listing approaches, Heuristics-based methods, Machine learning techniques and Feature-engineering methods respectively. More specifically, the taxonomy of existing malicious URL detection techniques is clearly indicated in Figure 2., which mentions all the four major groups of techniques. Further, Figure 2, also indicates the sub-category of URL listing techniques as well as Machine Learning algorithms in terms of secondary-level divisions with suitable methodologies.

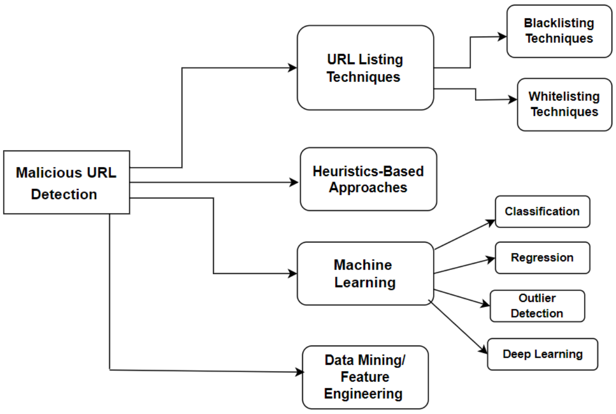


FIGURE 2. Taxonomy of malicious URL detection techniques.

B. PRISMA FLOW DIAGRAM

In this review article, the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) model [8] is employed, which facilitates the complete and transparent reporting of systematic reviews. In general, systematic reviews are essential for assessing the trustworthiness and applicability of review findings, which subsequently facilitates the research community to further replicate or update reviews. PRISMA reporting guidelines are designed to help the authors to prepare transparent reports of their reviews. Based on these aspects, the PRISMA flow diagram of article selection process, which is carried out in this review study is shown in Figure 3. At the onset of this review study, a total of 107 articles are collected from the online bibliographic databases, out of which 94 articles are selected for the review. Specifically, popular scientific databases, including IEEE Xplore, Elsevier ScienceDirect, SpringerLink, Wiley InterScience, Google Scholar, ACM and other electronic databases are used in this study. Further analysis is conducted to exclude the 40+ irrelevant articles, by means of identifying duplicates, out of scope and having inadequate information and so on as shown in Figure 3. Precisely, from the total of 94 chosen articles, 40+ were rejected by considering exclusion criteria's such as duplicates, out of scope and so on and subsequently the remaining 51 articles are mainly focused in this research study as depicted in Figure 3.

C. URL LISTING TECHNIQUES

In general, Listing techniques for malicious URLs detection, maintain a database of websites, which are previously identified as, either malicious (in case of Blacklisting techniques) or benign (in case of Whitelisting techniques), and are continuously updated by employing a combination of human verification and automatic mechanisms [9]. Specifically,

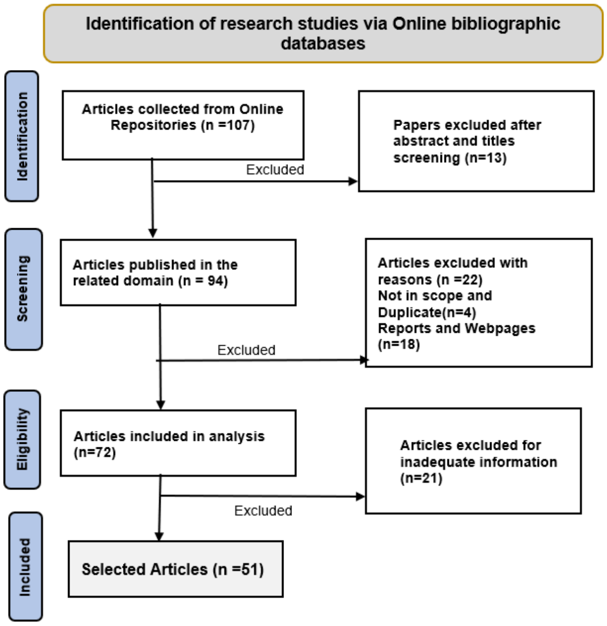

Identification of research studies via Online bibliographic databases
Identification
Screening
Eligibility
Included
Articles collected from Online Repositories (n =107)
Excluded
Papers excluded after abstract and titles screening (n=13)
Articles published in the related domain (n = 94)
Excluded
Articles excluded with reasons (n =22)
Not in scope and Duplicate(n=4)
Reports and Webpages (n=18)
Articles included in analysis (n=72)
Excluded
Articles excluded for inadequate information (n=21)
Selected Articles (n =51)

FIGURE 3. PRISMA Flow diagram of article selection process.

Blacklisting approaches detect and block URLs containing malicious software by maintaining a database of harmful URLs, while whitelisting techniques create a database of known safe URLs, permitting access only to those URLs [6], [9], [10], as described in the forthcoming subsections.

D. BLACKLISTING TECHNIQUES

Generally, Blacklisting techniques create and manage a database that contains a list of URLs, which are blocked from end-user systems due to their malicious content. For example, the authors presented a blacklisting technique in [11], which uses a restrictive list of websites, which are deemed dangerous due to their engagement with malicious content or suspicious behavior. There are several reasons for an URL to get blacklisted with one of the main causes, being the detection of malware on the webpage of a given website. For instance, an URL may be blacklisted due to the presence of a malware such as *Trojan horse*, which looks legitimate and gets added to the website's source code or installs malicious software onto the user's devices without their knowledge. Another common reason for blacklisting a webpage is, the presence of phishing sites, which disguise themselves as legitimate to deceive users into revealing their sensitive information. Further, the Websites involved in spam or hosting illegal contents are also blacklisted in order to follow the government policies and regulations. Furthermore, the websites that contain an excessive number of pop-ups or redirects may be blacklisted in order to mitigate potential risks.

There are multiple publicly accessible blacklists in the market such as PhishTank dataset [12] and URLhaus dataset [13], which get their URLs by the manual submissions from users or from external sources. After accepting URLs from users, they verify the respective URL by validating it against URL blacklisting services and various antivirus programs. Google Safe Browsing [14] and OpenPhish [15] are examples of blacklist services, which have developed their own analysis algorithms independent of external resources. Specifically, Google Safe Browsing [14] includes a dynamically updated database of websites, which are linked to malware or phishing activities. This blacklist service can be easily integrated with several browsers and applications. Another popular platform is PhishTank [12]- is operated by OpenDNS, an American DNS Resolution Service organization [16], which identifies and blocks malicious websites. Primary benefit of using PhishTank [12] is, its open-source nature, which allows users worldwide to report the suspected websites. These reports are reviewed by the community, and after verification, the phishing sites are added to the PhishTank blacklist [12]. PhishTank also provides APIs, that allow developers to incorporate phishing data into applications and thereby providing an additional layer of security solutions. Many browsers use PhishTank's blacklists to block access to sites, so that identity theft and financial loss can be prevented. In [17], the authors combined the listing methods with classification-based machine learning algorithms for detecting the malicious URLs. This approach utilizes a classifier model with four distinct layers comprising different filters, including blacklist and whitelist filters. Hong et al. [11] proposed a string-based blacklisting method along with machine learning models to detect the phishing URLs. The authors employed 18 different string features for representing each URL as a 19-dimensional vector, then it was inputted to ML models for the detection purposes.

The primary advantage of blacklisting approaches are their ease of implementation, which involves simple blocking of known harmful entities. Blacklisting techniques offer protection by blocking the currently identified threats with minimal efforts, when compared to the whitelisting approaches which require comprehensive implementation strategies. In general, blacklisting follows the permissive model, where access is open by default and only known threats are blocked. Users are not limited to the predefined set of websites/applications and can freely access most resources. Moreover, new threats can be swiftly added to the blacklist as they are detected.

E. WHITELISTING TECHNIQUES

Whitelisting techniques typically generate a list of approved websites, which are authorized for access by end-user systems. Precisely, whitelisting techniques are designed to deny access to every URL by default, unless it is explicitly added to the approved list of URL entries in the specified system. For instance, the authors proposed a whitelisting approach for malicious URL detection in [18], which employs a predefined permissive list of websites, such that URL entries not explicitly included in this list are automatically denied or blocked. By restricting access to only trusted entities and preventing exposure to harmful software or websites, whitelisting techniques create a controlled environment where only verified webpage entries are allowed to operate. On the other hand, a website may be whitelisted for several reasons: a) It may be recognized as a trusted site, such as well-known and reputable organizations; b) It may have undergone thorough assessments in terms of security and has been verified to be safe without any history of malicious activity or breaches and c) It may be adhering to industry-specific compliance standards and requirements.

Similar to blacklisting platforms, there are several publicly accessible Whitelist services, which are available on the Internet and the one, most widely used is *Apple's App Store* Whitelist. Specifically, Apple operates using a whitelist of applications on the App Store, which are verified to comply with the organization's strict security and privacy guidelines. This ensures that the users can download and use applications that are safe and harmless. Recently, Rafsanjani et al. [19] proposed a phishing detection technique using the multi-filter approach, which achieved 92.72% accuracy on a dataset comprising 1662 malicious and benign URLs respectively. Although this method is heuristic-based, it incorporates an auto-upgrade whitelist layer that helps to identify benign URLs and thereby eliminating the need for complex computations by excluding these URLs from additional considerations. One of the major benefits of using whitelisting techniques are, their secure nature, which allows to access only pre-approved entities in a certain system, while remaining websites are by default denied access. Thus, whitelisting blocks any unknown threats by default, and thereby reduces the risks of malware and phishing attacks. Whitelisting also helps to prevent zero-day attacks, which may not be recognized by traditional blacklists. Whitelists are generally easier to manage than blacklists, as they tend to be smaller and more focused. This reduces administrative overhead because, instead of maintaining extensive blacklists of potentially malicious entities, the whitelist only requires updation of trusted items, which makes it simple to manage.

F. HEURISTICS-BASED TECHNIQUES

Heuristics-based techniques are typically used to identify any potential malicious websites by analyzing patterns and structures that are commonly linked to malicious activity. Specifically, heuristics-based methods for malicious URL detection aim to detect the new and unknown threats by employing various rule-based analysis strategies. For instance, Nguyen et al. [20], introduced a heuristics-based malicious URL identification method, which involves creating a signature database of known attacks, sourced from antivirus or intrusion detection systems for verifying websites. Precisely, the authors compare the heuristic patterns and determine whether they match the signatures in the existing database or not for the detection purposes. Seifert et al. [21], presented a heuristics-based approach, which employs the extracted characteristic properties of existing phishing pages for detecting the new fraudulent webpages. Most of the heuristics-based techniques in the literatures [18], [19], and [20], analyze the syntax and structure of URLs to identify irregularities such as unusually long URLs, excessive sub-domains or hyphens, misspelled words in domain names and random strings in the path or query parameters. Further, Some heuristics-based methods consider the duration for which a domain in a URL is registered, as malicious sites are often created quickly and tend to have a short lifespan. Some of heuristics-based methods monitor the behavior of a website, after a user clicks on a link on that website. For example, the user may be redirected to multiple different domains, unauthorized files may be downloaded, malicious scripts could be executed and there may be malicious forms designed to collect sensitive data from the user.

In [20], Nguyen et al. introduced a heuristics-based technique, that utilizes 6 different URL features: PrimaryDomain, SubDomain, PathDomain, PageRank, AlexaRank [22] and AlexaReputation [23] respectively. In case of PrimaryDomain, the phishers use domains that are similar but include certain spelling errors to the original domain. The phishers may also add an arbitrary number of subdomains to the original domain. They may also use domains on the path subfolders to fool the users. PageRank is an algorithm that Google uses to determine the veracity of links. Usually, phishing websites have low Page Rank values, since these websites only exist for a short period of time. Similarly, AlexaRank is another site that ranks websites based on traffic information and access levels. AlexaRank value is low for websites that exist for only a short period of time. Further, AlexaReputation is a value that is calculated based on the number of links that are connected with the website under consideration from other websites. This value will be high in the case of phishing websites and low in case of legitimate websites. The authors calculated the heuristic value for each of these features followed by their weighted average in order to categorize the website as malicious or benign. If the value is positive, then the website is considered as a legitimate site, else it is considered as a phishing site. This technique [20] scored reasonably a good accuracy of 97% on a training dataset with 9661 phishing sites and a testing dataset with 1000 legitimate, 1000 phishing sites.

The research studies on heuristics-based techniques involving web crawling are introduced recently [18]. For instance, in [18], an extraction API is developed, which is embedded with scrapping techniques for the detection of malicious URLs. The API searches for form fields, downloadable contents, redirect links and constructs a vector based on the collected details. If this vector is already present in the collection, no further action is taken; otherwise, it only inserts the vector into the collection, and it is detected as a phishing site. Although, this technique achieves better accuracy, yet, scraping is proven to be computationally expensive, requiring high processing power. Similarly, the proposed API prototype requires a lot of improvements for dealing with URLs with elements, which cannot be easily interpreted by the API. The major benefit of heuristics-based approaches are, they are capable of handling zero-day attacks.

G. MACHINE LEARNING TECHNIQUES

Recently, Machine Learning (ML) techniques are gaining significant attention in the domain of malicious webpage identification, due to their ability to learn patterns from existing data and to predict the nature of unseen data [7], [10]. ML techniques typically use a set of URLs, where each URL is represented by a set of features. Based on these features, a prediction model is trained to classify the URLs as either benign or malicious [7]. The state-of-the-art literature on malicious webpage detection using ML techniques can be further divided into sub categories including Classification, Regression, Outlier detection and Deep learning techniques as shown in Figure 4. Specifically Figure 4 illustrates the four major categories of ML algorithms introduced in the literature of malicious webpage identification, which are detailed in the following subsections.

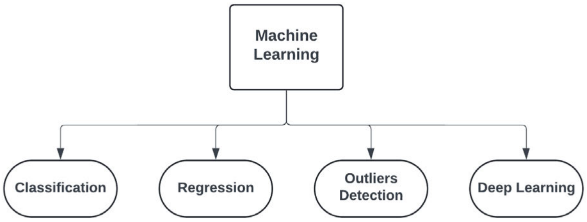


FIGURE 4. Machine Learning techniques with sub-categories introduced in malicious URL detection domain.

H. CLASSIFICATION TECHNIQUES

In general, classification is the process of developing a model that can differentiate between various classes and concepts, based on the analysis of training data with known class labels [10]. The purpose of any classification model is to predict the class labels of objects for which the classes are unknown in terms of categorical values. In the literature of malicious webpage identification, the most commonly used classification algorithms are Naive Bayes, K-Star, Linear Discriminant Analysis, Decision Tree, and Random Forest approaches, [7], [10], as shown in Figure 5. Additionally, Bagging and Boosting techniques, which are part of the ensemble learning category, have been introduced in the literature for the classification of malicious URLs, as indicated in Figure 5. Specifically, Figure 5 illustrates the various classification techniques proposed in the literature for detecting malicious URLs along with their corresponding reference frameworks.

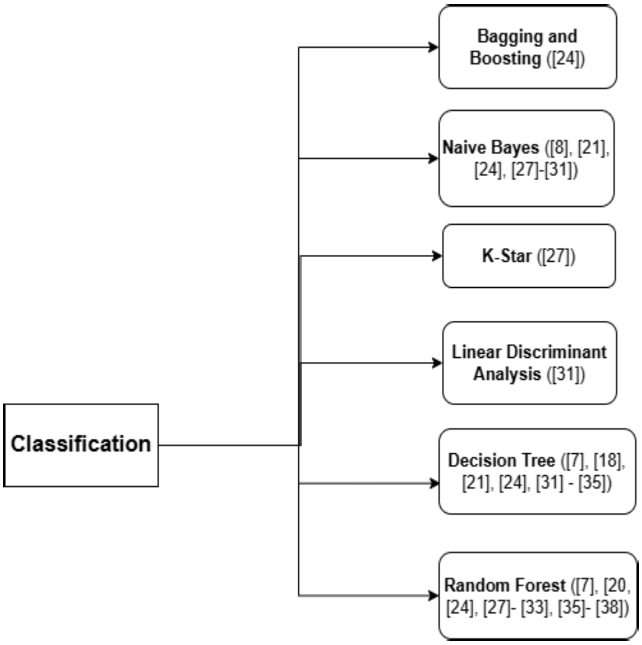

Classification
Bagging and Boosting ([24])
Naive Bayes ([8], [21], [24], [27]-[31])
K-Star ([27])
Linear Discriminant Analysis ([31])
Decision Tree ([7], [18], [21], [24], [31] - [35])
Random Forest ([7], [20, [24], [27]- [33], [35]- [38])

FIGURE 5. Classification algorithms with sub-categories employed for malicious URLs detection. (Note: Reference numbers in figure correspond to the related works cited in this article).

1) BAGGING AND BOOSTING APPROACHES

Bagging, which stands for Bootstrap Aggregation, is a ML strategy used to enhance the reliability and accuracy of predictive models [40]. Bagging typically uses the training data to generate multiple subsets by employing random sampling along with replacement. One of the main advantages of bagging is that it stabilizes predictive models by reducing variance, which minimizes the model's sensitivity to fluctuations in the dataset. Additionally, bagging enables parallel training of models, enhances efficiency, especially with larger datasets [40]. Boosting is a ML technique, that sequentially trains multiple weak learners, with each new learner focusing on correcting the errors of its predecessors [40]. Specifically, bagging trains the models in parallel and independently of each other whereas boosting builds the models sequentially, with each model trying to enhance the weaknesses of the previous models. One of the major advantages of Boosting is that, it improves the efficiency by correcting the errors sequentially. Boosting also addresses the under-fitting issues and thereby reduces the bias of the model, so that it can be applied to both the classification and regression tasks.

Manyumwa et al. [24] proposed a multi-class classification technique for detecting malicious URL attacks, by employing four different ensemble learners: XGBoost, AdaBoost, LightGBM and CatBoost. The proposed models are trained and tested simultaneously using the features from all four algorithms. Then, for a newly generated URL, each model provides its own prediction so that, the URL is classified into any one of the 4 categories: Benign, Malware, Spam and Phishing types. The authors used a dataset comprising 126,983 URLs collected from four different sources: DMOZ [25], PhishTank [12], URLhaus [13], and the WEBSPAM Dataset [26] and evaluated the performance of all four models.

2) NAIVE BAYES

Naive Bayes is a widely used classifier in the literature of malicious URL identification [8], [21], [24], [27], [28], [29], [30], [31], which is typically a supervised Machine Learning algorithm based on Bayes' Theorem [40]. Specifically, Bayes' theorem describes the relationship of conditional probabilities of statistical quantities in terms of defining the probability of an event based on prior knowledge of conditions that are related to that event. This classification technique assumes that, all the predictors are independent and allows the model to multiply each feature's probabilities to compute the likelihood of a class [28]. In general, the Bayes' Theorem is used to find the probability of an event occurring, given the probability of an event that has already occurred [40]. It is mathematically defined as follows:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{1}$$

where A is the event whose probability is to be calculated whereas B is the event that has already occurred. P($B|A$) is known as the likelihood probability, the probability that a hypothesis is true based on the evidence. P($A|B$) is known as the posterior probability, the probability of an event after evidence is seen.

3) K-STAR ALGORITHMS

K-star is an instance-based classifier, which is similar to k-Nearest Neighbors (kNN) algorithm [41]. However, unlike traditional kNN, which relies on simple distance measures to find nearest neighbors, K-star uses an entropy-based distance function [27]. K-star makes predictions for a new instance by comparing it, with the stored training instances and determines the nearest similar instance class. The advantages of K-star techniques are: a) less sensitive to noise and irrelevant features, b) effective management of missing instances and c) handling various different data distributions. For example, Ozgur et al. [27] employed seven different classification algorithms and three types of features including NLP features, Word Vector, Hybrid Features for detecting phishing URLs and compared the resultant performances. This framework using K-star algorithm, achieved an accuracy of 93.56% using NLP features, 81.05% with Word Vector features, and 95.25% with Hybrid Features, respectively.

4) LINEAR DISCRIMINANT ANALYSIS

Linear Discriminant Analysis (LDA) is utilized in the literature of malicious URL detection [31], as a supervised algorithm, primarily for multi-class classification tasks. In general, LDA technique projects a dataset of N dimensions onto a much smaller space with K, $K \leq N - 1$, without losing any class information. The advantage of the LDA is, it maximizes class separability, by enabling the identification of a feature combination that can distinguish between two or more classes [40]. LDA increases the distance between classes while simultaneously minimizing the spread within each class [40]. It also aids in dimensionality reduction by preserving the most important features needed in malicious URL detection. However, one major limitation of LDA is, its sensitivity to outliers, which can impact class separation. Additionally, LDA struggles when classes are not linearly separable, such that they cannot be perfectly divided by a straight line in two dimensions or a plane in three dimensions [31].

5) DECISION TREE

In the past few decades, Decision tree algorithms are widely used in the literature of malicious URL identification [7], [18], [21], [24], [31], [32], [33], [34], [35]. For example, Cao et al. [35], introduced a Forwarding-Based Malicious URLs detection system, which can be used in Online Social Networks. Decision Tree is typically a ML approach, represented as a tree-like structure, wherein the internal nodes represent a test on an attribute, branches represent outcomes of a test, and leaf nodes indicate the class labels [7], [40]. A decision tree is constructed by using a splitting criterion to choose the best feature for splitting at each level of the tree. This splitting process is repeated recursively until a stopping condition is met, typically when no further meaningful splits can be made [18]. The advantages of decision tree algorithms are, their ability to handle non-linear and complex relationships between the features and the target variable. They can also handle numerical and categorical data without making any assumptions regarding the data distributions [24], [40].

6) RANDOM FOREST

In the existing literature on malicious URL identification, Random Forest techniques are among the most extensively studied methods [7], [19], [23], [26], [27], [28], [29], [30], [31], [32], [34], [35], [36], [37], when compared to the other classification techniques. For instance, the authors presented a Random Forest-based detection in [27], in which they construct multiple decision trees on random subsets of the data. These trees are used to predict the class label of an unknown tuple, and the combined class label results are considered as the final class prediction. By combining the predictions of many decision trees, random forests enhance the prediction accuracy and variance, making it less prone to overfitting issues [32].

I. REGRESSION TECHNIQUES

Although regression techniques are commonly used for classification tasks in the ML domain, still there are relatively few studies in the literature on malicious URL detection using regression methods, which are described in this subsection. Specifically, sub categories such as Logistic Regression [20], [33], Classification and Regression Trees (CART) [43] and Ensemble Learning approaches [7], [24], [47] are widely used in the literature of harmful URLs identification in the past few decades, as indicated in Figure 6. More specifically,

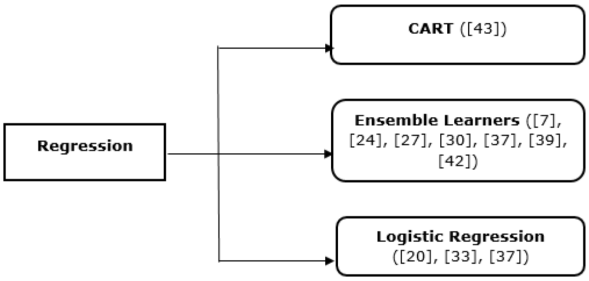

CART ([43])
Regression
Ensemble Learners ([7], [24], [27], [30], [37], [39], [42])
Logistic Regression ([20], [33], [37])

FIGURE 6. Regression techniques with sub-categories utilized in harmful webpages identification. (Note: Reference numbers in figure correspond to the related works cited in this article).

Figure 6 shows the widely popular Regression techniques used in the literature of malicious URLs detection along with the corresponding reference frameworks. Typically, Regression is a statistical method that determines the relationship between a dependent variable and an independent variable [40]. It determines the best-fit line and how the data points are dispersed around this line in terms of predicting continuous values.

1) LOGISTIC REGRESSION

In general, Logistic Regression (LR) is a binary classifier that uses the logistic (Sigmoid) function to determine the probability of an input being assigned a positive label [40]. If the output of the Sigmoid function is greater than 0.5, the result is classified as 1; otherwise, it is classified as 0 [33]. The sigmoid function is expressed as given by,

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

where $\sigma(x)$ represents the sigmoid function, x is the input variable and e represents the base of the natural logarithm [40], [41]. Several research studies are conducted in the recent years on malicious URL detection using LR techniques. For instance, Bannur et al. [33] utilized LR as a probabilistic model for malicious webpage identification, where the label is assigned by thresholding the calculated conditional probability. Further, the authors in [20] used logistic regression, which gives better accuracy compared to other classifiers and regression techniques. They used the dependent variable to decide whether a website is phishing, whereas the independent variable includes the proposed feature sets. Jain and Gupta [37] proposed a phishing detection system using LR technique, which employed hyperlinks found in HTML source as data features. Their model was trained on a labeled dataset of 2,544 websites, which comprised 1,428 phishing and 1,116 legitimate sites. It was evaluated using ten-fold cross-validation, and achieved a precision of 98.80% and F1-score of 98.59% respectively.

2) ENSEMBLE LEARNING

Ensemble Learners are specialized ML techniques, which combine the predictions of multiple individual models to create a more robust and accurate overall solution [40]. In general, in Ensemble Prediction, for the given M base learners, the ensemble combines their outputs as given by,

$$H(x) = \mathcal{C}\big(h_1(x), h_2(x), \ldots, h_M(x)\big) \tag{3}$$

where $C(.)$ is the combination functions such as regression or classification [41]. In the state-of-the-art literature of malicious URLs identification, several research studies are carried out using Ensemble learning techniques in the recent years [24], [27], [30], [37], [47]. For instance, Wang [30] introduced a malicious URL detection system that employs popular ensemble learning algorithms including LR, SVM, Random Forest, Gradient Boosting, and Bagging techniques. Recently, Catak et al. [39] employed popular Boosting algorithm, Gradient Boosting and ensemble learning for the detection of malicious URLs in the input environment.

3) AdaBoost

Adaptive Boosting, commonly known as AdaBoost, is an ensemble learning technique, which combines multiple weak learners, including single decision trees, for regression and classification tasks [40], [41]. In general, AdaBoost combines T weak classifiers $h_t(x)$ into a strong classifier as given by,

$$H(x) = \arg\max_{k\in\{1,\ldots,K\}} \sum_{t=1}^{T} \alpha_t\, \mathbf{1}\big(h_t(x) = k\big) \tag{4}$$

where $h_t(x)$, α_t indicate the prediction and weight of t-th weak classifier and k represents number of classes respectively. In the literature on malicious URL identification, AdaBoost is known for delivering highly accurate predictions due to its reliance on multiple weak classifiers [27]. Hou et al. [42] used AdaBoost algorithm combined with multiple decision trees for detecting harmful webpages, where each tree is weighted, to minimize the loss during training. This technique achieved an accuracy of 96.14% for multi-class classification, outperforming other methods such as Naive Bayes, SVM and simple Decision Trees.

4) LIGHT AND EXTREME GRADIENT BOOSTING

Light Gradient Boosting Machine, commonly known as *LightGBM*, is a gradient-boosting framework developed by Microsoft, that uses tree-based learning algorithms [41]. LightGBM has faster training speed, higher efficiency and can handle larger amounts of data. In the existing literature, gradient boosting is utilized for detecting malicious URL attacks and contributing to the fight against cybercrimes [24], [39]. For example, Ozgur Catak et al. [39] utilized Random Forest models and gradient boosting classifiers to create a URL classifier using URL string attributes as features for detecting the malicious URLs. Extreme Gradient Boosting, widely known as *XGBoost*, is a gradient-boosting framework, which employs a different tree-growing strategy in its implementation stages [41]. Specifically, LightGBM grows the trees leaf-wise by splitting the leaf with the largest gain, whereas XGBoost grows the trees level-wise, by growing each tree level evenly before moving on to the next level. Because of its level-wise tree growth, XGBoost is slower than LightGBM and consumes more memory.

Aljabri et al. [7] conducted a study on malicious URL classification using various models, including XGBoost, SVM, K-Means, kNN, and Convolutional Neural Networks (CNNs). The authors emphasize the superior accuracy of XGBoost, which outperforms the other models due to its capability to handle complex feature interactions and delivers high precision and robustness. Rafsanjani et al. [19] explored various ML techniques, emphasizing Gradient Boosting and Adaptive Boosting algorithms to improve the accuracy of malicious URL detection. They used a dataset of 5,000 real-world URLs collected from multiple phishing websites, including URLhaus and PhishTank. Jain and Gupta [37] utilized AdaBoost and six other algorithms for phishing detection and compared their performances. In their study, AdaBoost achieved an accuracy of 93.24% using NLP features, 74.74% on Word Vector features, and 92.53% on Hybrid features respectively. On the other hand, algorithms like Naive Bayes rely on simple probabilistic models, which perform well with small datasets but tend to result in higher false negative rates for more complex data. This makes AdaBoost a more robust choice for tasks where minimizing false negatives is crucial, such as in malicious URL detection problem. Manyumwa et al. [24] employed XGBoost, AdaBoost, LightGBM, and CatBoost for the classification of malicious URLs. In their study, XGBoost consistently outperformed AdaBoost and LightGBM in terms of precision, recall, and F1-score metrics. Additionally, XGBoost achieved higher accuracy, particularly in detecting long, abnormal phishing URLs and handling spam misclassifications.

5) CLASSIFICATION AND REGRESSION TREES

Classification and Regression Trees, popularly known as *CART*, is a ML algorithm, which uses decision trees for both classification and regression tasks [40], [41]. For classification, CART uses Information Gain (Gini Index) to identify the splitting criteria for each node, whereas for regression trees, it uses Mean Squared Error (MSE) and Mean Absolute Error (MAE) to decide the best split. For example, Yi et al. in [43] utilized logistic regression, CART, Back-Propagation based Neural Network and SVM to evaluate the effectiveness of various feature sets for detecting the malicious webpage. The authors employed four different feature sets including construction-based, IP-based, TTL-based and WHOIS-based features. In their experiments, logistic regression and CART were efficient in terms of computational speed but scored low accuracy. Backpropagation and SVM, on the other hand, achieved better accuracy but required longer training times due to their computational complexity. Additionally, it was found that, TTL-based features were strong indicators of malicious activity.

J. OUTLIERS DETECTION

In general, outlier detection is the process of identifying data points, that deviate from the normal distribution of a dataset and addressing them to prevent skewing of the data [40]. In ML algorithms, effectively handling outliers helps to prevent model bias, which is a critical factor in ensuring the performance of the given model. Specifically, Figure 7 illustrates the details of existing approaches that use anomaly detection and SVMs for identifying the malicious webpages.

1) ANOMALY DETECTION

A Phishing Detection and URL Checking framework using Web Crawling is introduced in [18], which detects the anomalies including malicious URLs and email contents. Wang [30], developed an effective system using feature extraction and ML algorithms to clearly identify and block the malicious URLs, which lead to phishing sites or malware downloads. In [35], the authors proposed a forwarding-based malicious URLs detection system, which analyzes the forwarding behavior of URLs and thereby identifies harmful ones in Online Social Networks.

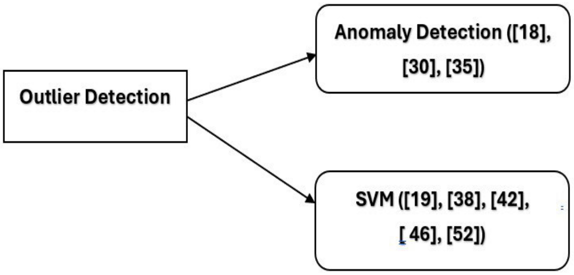

Outlier Detection
Anomaly Detection ([18], [30], [35])
SVM ([19], [38], [42], [46], [52])

FIGURE 7. Outliers detection techniques used for malicious URLs identification. (Note: Reference numbers in figure correspond to the related works cited in this article).

2) SUPPORT VECTOR MACHINES

Support Vector Machines, commonly known as *SVMs*, are machine learning algorithms widely used for outlier detection tasks [38], [40], [46], [50]. For example, Ahmed et al. [19] proposed a multi-layer filtering model to categorize websites by utilizing algorithms such as Naive Bayes, Decision Trees and SVMs. Specifically, the authors proposed an architecture with four different filters, each representing a distinct classifier. The first and second levels involve a stratification of a blacklist and a Naive Bayes filter. During the training process, the blacklist filter checks the blacklist and whitelist files to determine if the URL is present in either list. The next filter is a probabilistic filter, which calculates the probability of the URL being malicious (P1) versus normal (P2) using the Naive Bayes formula. When a URL

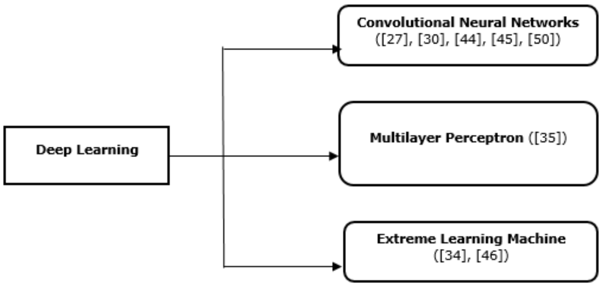

Deep Learning
Convolutional Neural Networks ([27], [30], [44], [45], [50])
Multilayer Perceptron ([35])
Extreme Learning Machine ([34], [46])

FIGURE 8. Deep learning algorithms used for harmful webpage detection. (Note: Reference numbers in figure correspond to the related works cited in this article).

is processed through the Naive Bayes filter, the calculated probability ratio is compared with a predefined threshold. If it exceeds the threshold, the URL is considered safe. Otherwise, it is sent to the next filtering layer- CART filter, which is a type of decision tree where each leaf node represents a classification decision (either malicious or normal). This multi-class filtering model achieved an accuracy of 79.55%, which is better than the simple Naive Bayes, single Decision Tree and SVM techniques.

Hou et al. in [42] tested four different classification algorithms including Naive Bayes, Decision Tree, SVM and Boosted Decision Tree for the detection of malicious URL attacks. The boosted decision tree was constructed by combining multiple decision trees into a strong classifier through a process called boosting, using AdaBoost algorithm. In this study, Boosted Decision Tree algorithm scored well, due to its ability to combine the strengths of multiple weak classifiers, thus reducing both bias and variances in predictions.

K. DEEP LEARNING

Deep learning is a ML approach based on Artificial Neural Network (ANN), where multiple layers of processing are employed to progressively extract higher-level features from the data [41]. Generally, Deep learning algorithms are multi-layered networks, that enhance the accuracy of predictions, by capturing the intricate patterns and dependencies within the data. The authors in [27], introduced a Phishing Attacks Detection framework by utilizing ML and Deep Learning Models. Most popular Deep learning techniques such as Convolutional Neural Networks (CNNs) [41], Multi-Layer Perceptrons (MLPs) [41] and Extreme Learning Machine (ELM) [41] are widely utilized in the literature of malicious URLs identification [17], [27], [35], [44], [50], as indicated in Figure 8. Specifically, Figure 8 shows the three popular deep learning methods widely used for identifying Malicious URLs in the literature along with the corresponding reference frameworks.

1) CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks, commonly known as *CNNs*, are a type of neural network models that closely mimic the human visual system and thereby extensively used for computer vision applications [41]. In general, for an input feature map X and a filter W, the convolution operation is defined as,

$$Y_{i,j}^{(k)} = \sigma\left(\sum_{m=1}^{M}\sum_{u=1}^{U}\sum_{v=1}^{V} W_{u,v}^{(k,m)} X_{i+u-1,j+v-1}^{(m)} + b^{(k)}\right) \quad (5)$$

where $Y^{(k)}$ indicates the output feature map for the k-th filter, $X^{(m)}$ indicates m-th channel of the input, $W^{(k,m)}$ represents the convolution kernel, $b^{(k)}$ is the bias term and σ indicates the activation function respectively. CNNs have more neural layers than traditional ANNs, as their structure includes an input layer, multiple convolutional and pooling layers, followed by an output layer. The input layer processes the raw image data and passes it to the convolutional layers, which perform convolutions by applying a set of kernels to the image, with each kernel detecting a specific feature [41]. The pooling layers, also part of the intermediate layers, reduce the spatial dimensions of the outputs from the convolutional layers using either max pooling or average pooling. Finally, the output layer provides the predicted class probabilities for classification tasks and continuous values for regression. Boyang Yu et al. in [45], introduced two categories of deep learning approaches: CNN-based and LSTM-based models for malicious URLs identification problem. CNN-based methods use convolutional techniques and vector embeddings to extract URL features. However, focusing exclusively on website features has limitations, as it overlooks other important factors like domain attributes and user behavior [45]. To address this issue, CNN methods can be combined with other approaches, such as incorporating DNS information to improve detection accuracy. Likewise, combining LSTM and CNN can further enhance the accuracy of the prediction framework.

2) MULTI-LAYER PERCEPTRON

Multilayer perceptron, commonly called as *MLP*, is a feed-forward artificial neural network that maps a set of input data onto a set of corresponding outputs [28], [41]. In general, the forward propagation of Layer l is represented as,

$$a^{(l)} = \sigma\left(W^{(l)} a^{(l-1)} + b^{(l)}\right) \quad (6)$$

where $a^{(0)}$ indicates x(input vector), $W^{(l)}$ & $b^{(l)}$ represent weights and biases and σ indicates the activation functions such as ReLU, sigmoid or tanh. Typically, MLP is represented as a directed graph consisting of an input layer, one or more hidden layers, and an output layer. Each layer can contain multiple neurons, and each layer is fully connected to the next. Except for the input layer, each neuron has an activation function and updates its value based on the values of the connected neurons and the weights of these connections. To train an MLP, the network is first initialized by specifying the number of layers and neurons and setting the initial weights and biases. During training, the input data is passed through the network via forward propagation, where each neuron applies its weights, biases, and activation functions to generate a prediction. These predictions are then compared to the actual target values using a loss function [35]. The resulting loss is used in backpropagation to calculate gradients and update the weights and biases. This process is repeated iteratively to minimize the loss and enhance the model's performance.

3) EXTREME LEARNING MACHINE

Altay et al. [34], presented a framework based on Extreme Learning Machine(ELM) technique, using single-hidden layer feed-forward neural networks to identify the malicious webpages. Unlike traditional neural networks, which adjust the weights iteratively, ELMs only optimize the weights connecting the hidden layer to the output layer. ELMs have only one hidden layer with randomly assigned weights and biases, which are kept fixed throughout the training process. For example, ALfouzan and Narmatha [46] highlight that, in the recent literature, deep learning is gaining prominence in detecting malicious URLs due to its ability to automatically extract useful features from the unstructured data.

L. FEATURE ENGINEERING

Li et al. [9] presented a new framework for improving malicious URL detections, by employing traditional classifiers including kNN and SVM combined with feature engineering methods, which outperformed deep learning methods. The authors also used non-linear transformation methods to extract the features that neural networks struggle to learn. In this study, 331622 URLs with 62 features were collected to validate the proposed feature engineering methods. The results showed that, the proposed methods significantly improved the efficiency and performance of certain classifiers, such as kNN, SVM and neural networks with the help of transformations methods.

M. METHODOLOGY-WISE DISTRIBUTION OF ARTICLES

Table 1 illustrates the distribution of the articles, which are considered in this review study along with the associated methodologies. Specifically, Table 1 summarizes the state-of-the art malicious URL detection techniques, including Whitelisting, Blacklisting, Heuristic approaches, Machine Learning and Feature Engineering, along with their respective subcategories, datasets, features, performance metrics and associated research studies.

N. QUANTITATIVE ANALYSIS OF PERFORMANCE METRICS

Table 2 illustrates the Summary of Quantitative Meta-Analysis of Performance Metrics of existing studies, which includes details of representative methods, datasets, performance metrics and trade-offs. Specifically, performance metrics such as Accuracy, F1-score, ROC-AUC of representative methods are highlighted in a summarized way, which will benefit the future studies in developing innovative solutions. Further, the detailed descriptions of Representative malicious URL detection methods along with their performance details and evaluation settings are also illustrated in Table 3 in the Appendix section of this article, for the benefit of future research studies.

III. CATEGORIZATION OF KEY-FEATURES USED

In general, various key-features of an URL are used in the literature of malicious URL identification, to differentiate between legitimate and harmful URLs, which are illustrated in the following subsections:

A. FULL URL-BASED FEATURES

In the literature, URL-based features are primarily used in the detection of malicious URLs [20]. URL-based features generally focus on the structure and content of the URL as shown in Figure 9. Specifically, Figure 9 describes the different parts of an URL including protocol, site name, domain name and path details. These features are considered as crucial, because the attackers can craft deceptive URLs that closely resemble legitimate ones. For example, Figure 10 displays a snapshot comparing fake URLs with legitimate ones and illustrates how attackers mimic the original URLs.

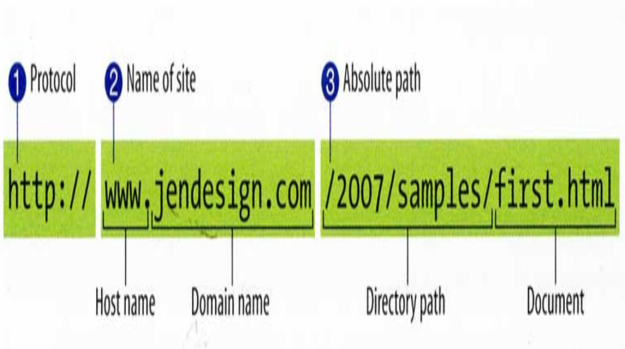


FIGURE 9. Different Parts of an URL.

Nguyen et al. [20] utilized URL-based features for detecting harmful URLs, by focusing primarily on different components of the URL including PrimaryDomain, SubDomain and PathDomain. By analyzing these elements, it is possible to identify the patterns, which indicate phishing attempts and thereby support more effective detection and prevention strategies. Specifically, Phishers often use slight misspellings or similar variations of legitimate primary domains, such as *'gooogle.com'* instead of *'google.com'*, as shown in Figure 10, since the attackers are unable to create the original domains. Further, Attackers may append subdomains to legitimate domains, which may mislead the users into thinking that they are on the real site. Attackers may also exploit the path or subfolder within the URL to fool the users by means of phishing attacks.

Vyas et al. [29] used PageRank, AlexaRank and AlexaReputation along with URL-based features for assessing the legitimacy of the URLs. These metrics provide additional

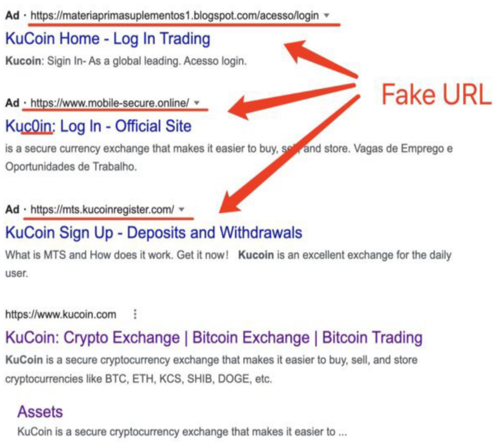
TABLE 1. Methodology-wise distribution of articles with details on datasets, features.

Methodology	Sub-category	Related Articles	Datasets Used	Features Employed	Main Performance Metrics
URL Listing	Blacklisting	[6],[9],[10],[11],[17],[46]	PhishTank (10000-12000 sample URLs)	URL, Strings	Accuracy
	Whitelisting	[6],[10],[18],[19],[46]	Trusted domain listings like Alexa	URL features	Accuracy
Heuristics-Based	Rule-based	[18],[20]-[23]	Custom, PhishTank	Handcrafted rules using URL length, symbols, IPs	Accuracy
Machine Learning	Classification	[7],[10],[18],[20]-[24], [27]-[35], [35]-[38]	Kaggle, PhishTank (10K to 1M+ URLs)	Features: lexical, host-based, content-based	Accuracy and F1-Score
	Regression	[7],[20],[24],[27],[30], [33],[37],[39],[42],[43], [46]	Kaggle (10K+ URLs)	Content-based, Lexical	RMSE and MAE
	Outlier detection	[18],[20],[27],[30], [35],[38],[46],[50],[51]	Custom Balanced URL sets	Unsupervised models like OCSVM	Precision, Recall, ROC-AUC.
	Deep Learning	[27],[30],[34],[35],[45],[50]	Larger datasets (100k-250k URLs)	CNN, RNN, LSTM on raw or embedded URLs	Accuracy
Feature Engineering	URL features	[9],[11],[29],[30],[43],[46]	PhishTank, DMOZ, Custom URL datasets	URL length, number of dots suspicious words, path tokens	Accuracy, TPR

TABLE 2. Summary of studies with quantitative analysis of performance metrics.

Methodology	Representative Models & Methods	Dataset/Size	Accuracy (%)	F1-score (%)	ROC-AUC	Remarks
URL Listing (Blacklisting)	PhishTank Matching [6],[9] -[11]	PhishTank (10k–12k)	85–95	–	–	Simple, interpretable; lacks detection of new attacks; high FP rate.
Heuristics-Based	Rule-based Filtering [20]-[23]	Custom/PhishTank	72–95	–	–	Lightweight and fast; limited adaptability to new threats.
Machine Learning (Classification)	Random Forest,SVM [20]-[24],[27]-[35]	Kaggle,PhishTank (10k–1M+)	90–97	90–97	0.95+	Balanced performance; interpretable; critical towards data and feature quality.
Machine Learning (Regression)	Logistic Regression [7],[20],[24],[27],[30]	Kaggle(10k+)	88–96	86–96	0.93–0.97	Used for risk scoring; well-suited for probability estimation.
Outlier Detection	Anomaly Detection, SVM [18],[20],[27],[30]	Balanced URL Sets	87–93	85–93	0.92–0.98	Effective for zero-day detection; provides unsupervised learning benefits.
Deep Learning	CNN, RNN, LSTM [27],[30],[34],[35]	100k–250k+ URLs	97–98.3	97–98	0.98+	Requires large datasets; best performance; higher computational cost.
Feature Engineering	Lexical,Host,Token Features [9],[11],[29],[30]	PhishTank, DMOZ, Custom URL datasets	82-93	–	–	Improves model accuracy; used in Preprocessing; cannot implement independently.

information regarding a website, such as the web traffic and the reputation of the website. Phishing sites typically have lower values for these metrics due to their short lifespan. Further, Wang [30] expanded the feature set by selecting 18 lexical features from URLs, including length, digit, special characters and other features. Specifically, length features include attributes such as the overall length of the URL, domain, path, and query. Digit features are important because malicious URLs typically contain more numeric characters than benign ones. Similarly, malicious URLs often have a higher frequency of special characters, including dots, slashes, dashes, percentage symbols, and underscores. Other



Ad · https://materiaprimasuplementos1.blogspot.com/acesso/login ▾

KuCoin Home - Log In Trading

Kucoin: Sigin In- As a global leading. Acesso login.

Ad · https://www.mobile-secure.online/ ▾

Kuc0in: Log In - Official Site

is a secure currency exchange that makes it easier to buy, sell, and store. Vagas de Emprego e Oportunidades de Trabalho.

Ad · https://mts.kucoinregister.com/ ▾

KuCoin Sign Up - Deposits and Withdrawals

What is MTS and How does it work. Get it now! **Kucoin** is an excellent exchange for the daily user.

https://www.kucoin.com ⋮

KuCoin: Crypto Exchange | Bitcoin Exchange | Bitcoin Trading

KuCoin is a secure cryptocurrency exchange that makes it easier to buy, sell, and store cryptocurrencies like BTC, ETH, KCS, SHIB, DOGE, etc.

Assets

KuCoin is a secure cryptocurrency exchange that makes it easier to ...

FIGURE 10. **Snapshot: Fake URLs Vs Legitimate URLs.**

notable features include the presence of an IP address and the occurrence of sensitive words (such as "account," "banking," and "login"), which are commonly found in malicious URLs.

1) URL DOMAIN PORTION-FOCUSED FEATURES

Domain-based features are generally a subset of URL-based features, which mainly focus on domain portion of the URL. For example, Hameed et al. [17] leveraged domain-based features for the detection of malicious URLs, in which the authors analyzed different characteristics of domain names. Specifically, some of the domain-related features considered in this study are: a) Presence of more than four consecutive numbers in the domain name, which indicate randomness in the creation of the URL; b) Presence of the special characters, such as '#', '$', and '_', which are common traits in suspicious domains; c) Domain name belonging to common top-level domains, such as ".com", ".en", ".net", and ".org"; d) The count of periods in the domain name, which is an indication of the domain complexity, the number of subdomains utilized and e) The total length of the domain name, since excessively long or short domain names can raise suspicions.

Li et al. [9] employed different domain-based features such as expiration days, register days, domain and its sections for detecting the phishing URL attempts. Specifically, expiration days refers to the number of days remaining before the domain expires. Phishing websites are created quickly and are taken down shortly after they are detected. Attackers register domains for only short periods to avoid detection, which can be reflected in the expiry days count [43]. Similarly, register days indicate the number of days since the domain was registered. Older domains are usually considered more trustworthy. Newly registered domains are considered suspicious since phishers often register domains shortly before launching their attacks. The domain and its sections include the server section and the domain section. In malicious websites, the server sections tend to be much longer than those of benign websites. One of the primary advantages of domain-based features, are their simplicity, since they are easy to extract from the URLs. Features such as the length of the domain name or the number of special characters can be quickly calculated and interpreted. These features can be computed in real time and require minimal computational resources and thereby indicate their suitability for real-time detection systems.

B. LEXICAL FEATURES

Lexical features are extracted based on the premise that malicious websites exhibit distinct characteristics, which result in recognizable patterns within their URL text [11]. In many instances, users can spot suspicious URLs just by examining their appearance. This is particularly true for phishing URLs, where the URL closely mimics that of the legitimate page it tries to imitate. Lexical features refer to all textual aspects of the URL itself, excluding the content of the website it directs to. These features are appealing for detection due to their fast processing time, minimal data storage requirements, and no need to access other services for extraction. Common detection methods typically treat the lexical features of the host name and path separately due to their differing impacts on classification [11]. Traditional lexical features encompass commonly used attributes such as the length of the host name, the total length of the URL, and the frequencies of characters within the URL. Additionally, the presence or order of tokens in the host name (separated by '.') and in the path (separated by '.', '/', '?', '=', '-', and ' ') are also considered.

C. CONTENT-BASED FEATURES

In general, content-based features involve analyzing the content of the entire web page itself. For example, Xuan et al. [36] defined a malicious URL detection method using content-based features, which are extracted from the full download of a web page. The authors highlight that, the main sources of these features are the HTML content of the web page and any JavaScript used. These features provide a deeper understanding of the site's behavior and intent by analyzing elements such as links, forms, and scripts. Aljabri et al. [7] utilized content-based features such as Request URL, URL of Anchor, Links in Tags, and SFH features for malicious URL detection. The Request URL feature checks whether the web page contains any external objects, such as images or videos, that are loaded from a different external domain. This can be an indication of a phishing attack because malicious sites often load content from external domains to avoid detection. URL of Anchor feature examines the URL anchor that links to a web page, that is, the URLs that are used in anchor tags ($< a >$). A high percentage of anchor links point to external domains, can be an indication of malicious activities. Links in Tags analyzes the presence of $< meta >$, $< script >$, $< link >$ tags since they can contain links or code that can be used to track or manipulate the users.

D. OTHER FEATURES

In addition to lexical, domain and content-based features, there are several other features in the literature, which are utilized for malicious URL detection. For example, Yi et al. [43] used IP-based features, TTL-based features, as well as WHOIS-based features, along with Lexical features. IP-based features include features such as the number of IP addresses, which indicate how many unique IP addresses are associated with a domain and number of countries. Phishing websites typically change their IP addresses to evade detection, so a high number of IP addresses is an indication of malicious activity. The distribution of the IP addresses across multiple countries can indicate a phishing site since legitimate websites have a more concentrated presence in specific regions. TTL (Time-To-Live) refers to the lifespan or duration of a particular DNS record, indicated as the average and standard deviation of TTL values. The average TTL can help to assess the stability and reliability of a domain, as malicious websites typically have shorter TTL values, indicating frequent changes. A high standard deviation in the TTL value suggests inconsistency in how long DNS records are cached. WHOIS-based features include the domain's lifetime and active time. The lifetime of a domain refers to the duration between its registration and expiration dates, with short-lived domains often being associated with phishing attacks. The active time of a domain indicates how long it has been operational, with phishing sites typically having shorter active times.

Manyumwa et al. [24] proposed three additional types of features, which are word-based features, Shannon Entropy features, and KL Divergence features. Word-based features identify the presence of specific words that indicate the malicious intent. Some of these can be suspicious words, such as *login*, *online*, *verify*, *secure*, and *submit*. These terms are often used by attackers to deceive users into thinking that they are interacting with legitimate sites. There also exist obfuscated words, where the attackers create misspellings or variations of well-known brand names, such as '*paypell*' instead of '*Paypal*. The authors also employed Shannon Entropy to evaluate the randomness within URL strings. Malicious URLs typically exhibit higher entropy compared to benign ones. Similarly, KL divergence can also be used to calculate the entropy. Generally, human-constructed URLs tend to include sensible words and a reasonable mix of characters, while digitally generated URLs are more random and consist of meaningless strings.

IV. POPULAR EXPERIMENTAL DATASETS

In the literature of malicious URL detection, multiple research frameworks employed a few popular publicly-available datasets for experimentation purposes, which are listed below:

1) PhishTank [12]: One of the most commonly used databases for phishing identification purposes.
2) DMOZ [25]: Also known as the *Open Directory Project*, which consists of a directory of human-reviewed websites.
3) UCI Machine Learning Repository [47]: This repository contains popular datasets, that are employed in multiple ML domains such as health care, bio-informatics and cybersecurity.
4) Alexa Whitelists dataset [49]: This dataset provides a list of the most popular and frequently visited websites.
5) OpenPhish dataset [15]: The most commonly used dataset for the detection of harmful URLs, which offers free and paid access.
6) URLhaus dataset [13]: This provides plain-text based malware URL lists, which are generated and updated for every 5 minutes.
7) WEBSPAM dataset [26]: This includes multiple versions of popular datasets including webspam-UK2006 and webspam-UK2007 datasets.

The detailed descriptions of these datasets are also provided in the Appendix section of this article, for the benefit of future research studies.

V. CHALLENGES AND DISCUSSION

This section outlines key research directions within each category of malicious URL detection techniques, which remain unexplored and present ongoing challenges in the field. prioritization of research challenges by considering their critical aspects, could enhance the clarity of this section. However, the malicious URL detection field is highly dynamic in nature, due to which the relative importance of specific challenges may vary depending upon their application scenarios and emerging technological trends. Due to this aspect, we presented a broad roadmap of research directions, which covers each category of techniques and thereby leaves prioritization open to domain-specific applications and future studies.

A. BLACKLISTING TECHNIQUES: RESEARCH DIRECTIONS

Using blacklisting techniques for the identification of malicious URLs has the advantages of being fast and easily implementable, with a low false positive rate. However, few key directions for future research are given by:

- Blacklisting techniques can only address known threats, which leaves the systems vulnerable to the newly emerging threats, that are not yet identified or added to the blacklist database.
- Maintaining an updated blacklist database is essential for better accuracy of the URL detection system, which is more complex and resource intensive due to the need for dynamic updates.
- Generally, blacklisting techniques fail to provide robust protection against *zero-hour attacks*, as they classify only 47–83% of new phishing URLs in a 12-hour

period. Thus, these methods can be easily bypassed using obfuscation methods.

- Sometimes, blacklisting may lead to higher rate of false positives and block legitimate websites, since the list never becomes exhaustive.
- Although blacklisting techniques are effective, the attackers can sometimes easily bypass the system by making changes to one or more components of the URL string.

To address these issues, blacklisting could be used as an initial step in detecting malicious webpage, rather than relying on it as the sole identification method. The frameworks which integrate blacklisting with more advanced techniques, remains largely unexplored in this research domain.

B. WHITELISTING TECHNIQUES: RESEARCH DIRECTIONS

While whitelisting approaches offer significant benefits in terms of security towards harmful URLs identification; yet, they present certain challenges that need to be addressed:

- Whitelisting techniques sometimes may end up with compatibility issues. Precisely, if a certain application or website is not added to the whitelist in a timely manner, then there might be disruptions in accessing the service by the users.
- Whitelisting approaches often impose constraints on the user's ability to access websites, and it may require continuous dynamic updates, in order to ensure that all the trusted sites are included [17].
- There might be the chances of false negatives; that is, some legitimate websites might be inadvertently excluded from the whitelist, which may lead to potential attacks on the service by malware or phishing.

C. HEURISTICS APPROACHES: RESEARCH DIRECTIONS

One of the main advantages of heuristics-based techniques is that, they are effective against *zero-day threats*, such as newly created phishing URLs, that have not yet been reported. Further, heuristics approaches are capable of analyzing URLs, as soon as they are encountered and thereby provides dynamic real-time protection against malicious activities. However, few of the research issues of heuristics-based techniques are yet to be resolved:

- If the signature database generated by the heuristics-based system is not updated dynamically, then it may result in zero-day exploits.
- Heuristics-based systems, may misclassify URLs as malicious when they contain characteristics, which are usually present in harmful URLs.

D. CLASSIFICATION TECHNIQUES: RESEARCH DIRECTIONS

Although, several classification algorithms are popularly employed in the literature of malicious webpage detection, yet, few of the research issues need to be explored, which are given by:

- K-star techniques can be computationally intensive, as it need to compare every new instance with every stored training instance using complex calculations.
- K-Star approaches are often memory-intensive because they require storing all the training data. As an instance-based learner, this can lead to reduced performance when dealing with larger datasets.
- Though, Linear Discriminant Analysis (LDA) supports dimensionality reduction by preserving the essential features; yet, it is more sensitive to the outliers, which may affect the class separation performances.
- The performance of LDA in identifying malicious URLs may be suboptimal if the classes are not linearly separable. For the model to perform well, the classes must be perfectly separable by a straight line in two dimensions or a plane in three dimensions.
- The decision tree techniques are widely used in the literature of malicious URL identification due to their ability of handling non-linear and complex relationships between the features and the target variable. However, decision tree techniques suffer due to overfitting issues and sensitivity towards small variations. Specifically, they overfit the training data, especially when the depth of tree is high. They result in different splits and tree structures, even when there are small changes in the dataset.
- In Random Forest approaches, while averaging multiple trees helps to reduce the risk of overfitting, still construction of these trees and aggregation of results demands more computational power and memory. Additionally, training Random Forests can be time-consuming, as the training time increases with the number of trees in the forest.

E. REGRESSION TECHNIQUES: RESEARCH DIRECTIONS

Although regression techniques are widely utilized for identifying harmful webpages, few of the research challenges, that are yet to be explored towards this category of techniques are given by:

- Although, Logistic Regression is capable of capturing complex relationships in the linear data; yet, more sophisticated models are needed for capturing non-linear data.
- To deal with multi-collinearity issues, variance inflation factors must be utilized, so that highly correlated variables can be removed using regularization techniques.

- Logistic regression techniques are more sensitive to outliers, which can be addressed by means of utilizing sensitivity analysis and robust transformation methods.
- While Bagging enables parallel training of models and improves accuracy for larger datasets, it also leads to a loss of model interpretability. Specifically, if the proper procedures are not followed, the resulting model may suffer from significant bias.
- Although, Bagging techniques tend to provide highly accurate results; still, they suffer due to their computationally-expensive nature, which in turn discourages its usage in certain instances.
- AdaBoost algorithms provide accurate results; yet, they are very sensitive to noise and outliers, since it places higher weights on the misclassified data points, which leads to overfitting issues.
- Although in LightGBM training duration is faster; yet, there can be slow prediction times due to the complexity of the decision trees, which becomes a problem for some of the real-time applications.
- LightGBM provides higher accuracy, but it suffers due to overfitting on small data samples, which can be resolved only by employing careful hyper-parameter tuning strategies.
- The XGBoost algorithm is capable of handling complex feature interactions, but it requires significant computational resources and time, especially while dealing with large datasets, which makes it unsuitable for resource-constrained environments.

F. OUTLIERS DETECTION: RESEARCH DIRECTIONS

Though SVM techniques are quite popularly employed in the literature of malicious webpages identification; yet, they face several research challenges that require further exploration, as outlined below:

- Selection of good Kernel function is the trickiest part, which mainly decides the performance of an SVM-based model in the malicious webpage classification task.
- SVM may consume longer training time, when compared with other classification models for larger datasets, which potentially leads to degraded performance.
- Fine-tuning critical hyper parameters of SVM, such as Gamma, is relatively complex because visualizing their impact on the model's performance is challenging.

G. DEEP LEARNING ALGORITHMS: RESEARCH DIRECTIONS

Although, Deep learning techniques such as CNNs and MLPs are widely used in several research studies for identifying harmful URLs; yet, few of the research issues that need to be addressed are given by:

- Deep learning models provide promising results when compared to other ML techniques. However, they struggle with evolving malicious websites due to the inherent *black-box* nature of these models. Further, the black-box nature of deep learning methods limits their interpretability, making it difficult to understand and optimize the feature selections.
- Handling the issues of false positives and false negatives, still remains a challenge when using Deep learning models for larger-sized datasets.
- Deep learning algorithms are computationally intensive and require GPUs for training process. Additionally, retraining the models is necessary when new data becomes available, which adds further complexity to the training process.
- The reliability and accuracy of Deep learning-based detection methods heavily depend on the quality of the dataset used, which it turn creates more constraints on the usage of suitable datasets [3].

H. KEY-URL FEATURES: RESEARCH DIRECTIONS

- Although domain-based URL features are easy to implement, they are limited in scope and may struggle to detect various types of malicious URLs.
- Many malicious URLs may consists of legitimate domain structures, which makes it difficult to classify them solely based on these features. As a result, attackers can easily evade detection by using domains that appear legitimate, such as well-known top-level domains.
- Domain-based features fail to consider the web page content or user behavior. As a result, they may fail to detect URLs, which lack apparent domain-based anomalies but still contain harmful content.
- Content-based feature extraction requires significant computational resources due to the volume of information that must be processed. They also raises security concerns regarding access to potentially malicious webpages.

VI. OTHER EMERGING METHODS

This section highlights some of the emerging techniques in this malicious URL detection domain, which are currently gaining traction and showing potential in research applications, as described as follows:

A. HYBRID DEEP LEARNING ARCHITECTURES

A prominent trend in this malicious URL identification domain is, the usage of hybrid deep learning architectures, which combine deep learning models such as CNNs and Recurrent units. For instance, in 2025, Egigogo et al. [52] developed a CNN-Bidirectional Gated Recurrent Units (BiGRU) Hybrid Model based framework for the detection of website phishing attacks. In their system, parallel convolutional layers extract the high-level n-gram features, whereas BiGRU capture contextual sequences in URLs. This hybrid deep learning architecture processes raw URLs directly and eliminates manual feature engineering and thereby achieves an accuracy of 98% on large-scale datasets. Very recently, different hybrid architectures are introduced in this field, which combine convolutional feature encoders with Long Short-Term Memory (LSTM) modules for achieving robust temporal and structural understanding [53], [54]. Latest studies using hybrid models, incorporated attention mechanisms into LSTM or GRU models, which enhanced the recognition of key URL segments and thereby improved detection accuracies [55], [56], [57].

B. GAN BASED METHODS

Recently, Generative Adversarial Networks (GANs) based methods are gaining interest in this domain, since they enhance the robustness of detection models by exposing them to challenging and diverse scenarios. Adversarial learning such as GANs are effectively utilized in the recent years, for generating realistic malicious URLs, improving robustness and enhancing generalization capabilities of detection systems [58], [59], [60]. Specifically, these models demonstrated improvements in recall and F1-scores, when integrated into adversarial training pipelines for phishing URL detection. In addition, techniques such as Synthetic Minority Oversampling Technique (SMOTE) [61] are applied to address the class imbalance related challenges in real-world URL datasets. Recent studies indicate that, SMOTE-enhanced datasets, when combined with deep neural classifiers achieved significant improvements in both precision and recall scores [59].

C. FEATURE REPRESENTATION AND ENSEMBLE LEARNING

In this harmful webpage identification domain, recent studies emphasize the integration of optimized feature representation with Ensemble learning approaches for enhancing the detection results. Specifically, Ensemble techniques, which combine conventional machine learning techniques such as RF, SVM and XGBoost with Deep neural networks achieve a clear balance between interpretability and effectiveness. These Ensemble-based frameworks proved their strong capabilities in terms of handling multi-class URL classification tasks comprising benign, phishing, and malware URLs. For instance, recently in 2025, Raja et al. [62] introduced a XGBoost and ANN based Ensemble Learning framework, which reported an accuracy of 96.7% and F1-score of 95.0% for the malicious URL detection task. In [63], CatBoost ensemble method is incorporated with BERT-derived URL embeddings and metaheuristic-selected features for obtaining better accuracy and F1-scores, which are 97.1% and 96% respectively.

D. GNN-BASED TECHNIQUES

Recently, Deep Learning frameworks based on Graph Neural Networks (GNNs) are scoring better results in cyber security applications such as Phishing websites detection. For instance, Bilot et al. [64], presented a phishing websites detection framework, which utilized a Deep Learning architecture based on GNNs along with the hyperlink graph structure of websites. In [65], graph based reasoning method is employed along with the message passing mechanism of GNN for identifying phishing URLs.

E. TRANSFORMER-BASED MODELS

In the recent years, transformer architectures based methods are gaining attention in the malicious URL detection domain. For example, Asiri et al. [66] presented a PhishTransformer framework, which combines CNNs with transformer architectures to extract features from website URLs, so that the phishing and legitimate websites can be clearly distinguished. Though this framework scores promising results, yet input pipeline size of this model is quite huge. Very recently, hybrid models employing transformers and Deep learning architectures are gaining popularity in this domain. For example, Nguyet et al. [67] proposed an integrated model using DL classifiers and pre-trained transformer for identifying the phishing webpages. Specifically, the authors utilized a Residual Network (ResNet) to learn feature representations of website URLs, Temporal Convolutional Network (TCN) to eliminate irrelevant features and Masked and Permuted Pre-training for Language Understanding (MPNet) to handle language modelling issues. In [68], a new deep learning based phishing detection system is presented, which integrates the strength of Variational Autoencoders (VAE) and deep neural networks (DNN) for fake URLs detection problem.

F. EVOLUTION OF MALICIOUS URL DETECTION: RESEARCH ROADMAP

Figure 11 illustrates the evolution of malicious URL detection literature starting from URL listing methods to emerging techniques such as transformer-based models by means of a Progression diagram. Specifically, early URL listing methods utilized static lists of trusted/whitelisted and blocked/blacklisted URLs. Although, these listing techniques are simple to implement, yet they are less effective towards new threats. Heuristics-based approaches employed manually crafted flexible rules, still they are limited towards the detection of unknown attacks. Machine learning techniques such as classification utilized trained models for accurately identifying harmful webpages, whereas Feature engineering methods employed URL-based features such as lexical for their detection purposes. Figure 11 highlights the emerging methods of this domain, which include: a) Hybrid Deep Learning architectures, which combine multiple deep learning models; b) GAN-based Models, which generate adversarial examples for strengthening the models; c) GNN-based techniques, which utilize graph neural networks for analyzing the URL relationships; d) Feature Representation & Ensemble Learning approaches, which enhance robustness by combining different models; e) Transformer-based Models, which employ attention mechanisms for achieving better detection accuracy. In this way, this research roadmap provides a structured overview of the research landscape and helps the future researchers to understand potential future directions.

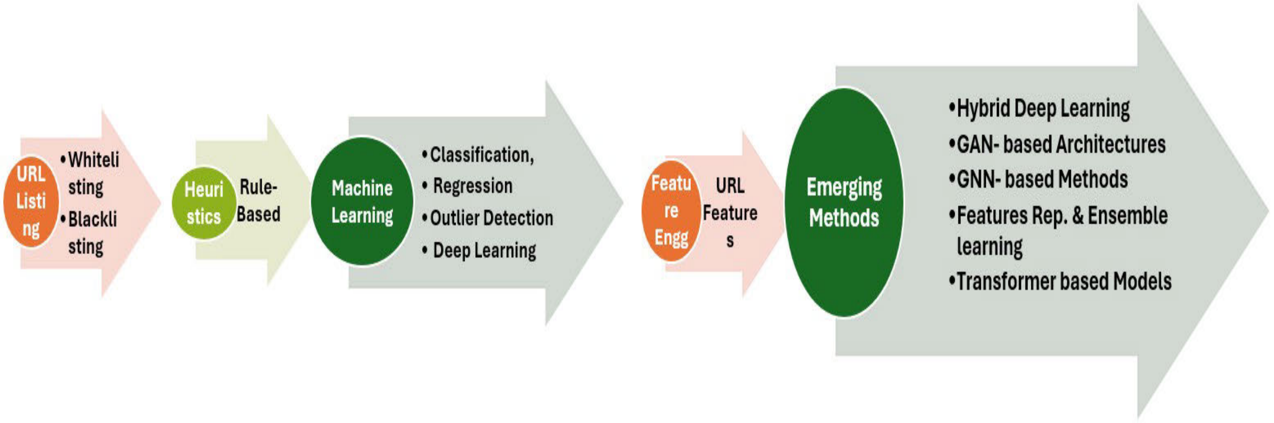


FIGURE 11. Malicious URL Detection Domain - Research Roadmap.

VII. THEORETICAL, MANAGERIAL IMPLICATIONS AND REAL-WORLD DEPLOYMENT CHALLENGES

A. THEORETICAL IMPLICATIONS

Due to the growing dependency on digital platforms, which is accelerated by the COVID-19 pandemic, the access to information is tremendously increased and thereby introduced greater vulnerability to cyber threats. Typically the increased rate of cybercrimes such as phishing attacks, emphasize that more advanced detection techniques are very much essential for protecting the end-users from cyber attacks. Based on these aspects, this review suggests a few of the theoretical implications in multiple areas as given by:

- **Technology Enhancements:** Due to the growing usage of digital platforms, the end-users are more susceptible to cyber attacks, which in turn emphasizes more on re-design/ re-modeling combined with re-evaluation of the existing cyber-security frameworks. Specifically, the theoretical models of digital security especially those handling URL-based attacks, must be strengthened to address the growing threat scales and their complexities.
- **Multi-faceted Detection Techniques:** This review study presents a comprehensive taxonomy of detection techniques including Listing, Heuristics, Machine Learning, and Feature Engineering approaches and thereby contributes to the theoretical development of this domain. However, this review suggests that no single technique is sufficient for the efficient detection of all kinds of harmful URL attacks. Therefore, it gives emphasis on the integration of diverse existing techniques, by considering their different nature of features and varied datasets.
- **Innovative Detection Techniques:** In this review, key research challenges related to different detection techniques and feature types are identified, which encourages the development of innovative URL-detection approaches to address the various cyber security challenges.
- **Interdisciplinary Natured Approaches:** The interdisciplinary nature of this review study, combines the elements of computer science, machine learning and cyber security and thereby reflects the popular trend towards cross-disciplinary solutions. In this aspect, the cyber defense and security risk management must be explored in terms of integrating insights from AI, data science, and behavioral studies, to address the complexity of modern cyber threats.

B. MANAGERIAL IMPLICATIONS

The rapid growth of digital platform usage, which is driven by technological advancements as well as the COVID-19 pandemic, enforces the critical need for businesses and organizations to implement robust strategies to protect against a wide-range of cyber attacks. Based on these aspects, the managerial implications derived from this review study are highlighted as follows:

- **Prioritizing Cyber security Measures:** In response to the growing number of data breaches, the organizations need to implement preventive measures such as more-advanced URL filtering tools to minimize vulnerabilities and to protect sensitive data.
- **Adopting Multi-layered Detection:** Organizations should focus on implementing the multi-layered approaches to the malicious URL detection problem, which integrates more comprehensive security strategies.
- **Implementing Advanced Threat Intelligence:** The review article highlights the importance of early-detection of malicious URLs to protect users from wide-range of cyber threats. In this aspect, the organizations must invest in advanced threat intelligence platforms which can provide real-time updates on emerging threats by analyzing the URLs for potential dangers and thereby enable quicker responses to prevent data breaches.
- **Enhancing Data Security Measures:** Since phishing attacks account for the major portion of data breaches, organizations must consider strengthening the security of sensitive data in terms of Multi-factor authentication

(MFA) strategies, encryption techniques and other robust data protection measures.

C. REAL-WORLD DEPLOYMENT CHALLENGES

This review study emphasizes few of the real-world deployment challenges, which are yet to be focused in this malicious URL identification domain, as given by:

- **Need for Evolving Threat Landscape:** The complexity of URL-based attacks is continuously increasing and hence the state-of-the-art cybersecurity frameworks need to frequently redesign and re-evaluate their strategies in order to handle these challenges.
- **Requirement of Universally Effective Solutions:** The existing frameworks, require universal solutions, which can effectively handle all types of malicious URL attacks by integrating hybrid detection mechanisms.
- **Complicated Integration Issues:** Although, integrating diverse detection techniques and multiple datasets for the malicious URL detection problem enhances robustness, still it is quite challenging due to factors such as different feature types, data formats and detection concepts.
- **Cross-Disciplinary Requirements:** Although, effective cyber protection can be achieved by integrating methodologies from different domains including AI, data science, and behavioral pattern analysis, still it poses collaboration as well as implementation challenges.

VIII. CONCLUSION

This review article presents an extensive literature review by means of highlighting the taxonomy of all the malicious URL detection techniques by considering the limitations in the literature, different features types and the datasets used, when compared to the existing single-technique specific research reviews. As a result of this analysis, it also highlights the research challenges of every category of the detection techniques, its implications and thus encourages the new researchers to come up with possible solutions.

In future, this review article could be expanded to provide a detailed analysis of each category of URL features, their corresponding datasets, and the limitations associated with them, which are yet to be thoroughly explored in the literature. While the datasets predominantly contain phishing URLs, this composition aligns with real-world reporting trends, where malicious URLs remain the most common cyber attacks. From another perspective, open-access, well-labeled datasets for non-phishing threats such as malware distribution and C2 infrastructures remain scarce, which inevitably leads to phishing-dominated benchmarks. As a part of future work of this study, we are planning to incorporate datasets with broader coverage of malware and C2 infrastructures for enhancing generalization capabilities.

This review illustrates techniques, whose objective is only to classify quickly, whether the given URL is safe or malicious with minimal risk by employing URL-based features, without necessarily visiting the webpage. In contrast, Deep crawling identifies malicious payloads/behaviors by fetching, executing page contents and examining the actual content, behavior of web page linked to the URL with the help of full webpage resources (like Images, HTML, Javascript, hidden elements). In future work, this study can be extended to add discussions on implementation of sandboxing, security isolation, and its data handling issues.

IX. DECLARATIONS

A. AVAILABILITY OF DATA AND MATERIALS

No new datasets were generated or analyzed during the current study.

B. COMPETING INTERESTS

The authors declare no competing interests.

C. FUNDING

Not applicable

D. AUTHORS' CONTRIBUTIONS

All authors contributed equally.

APPENDIX

E. BENCHMARK OPEN-ACCESS DATASETS

The most widely used publicly available datasets in the domain of malicious webpage detection are presented as follows:

1) PhishTank

PhishTank [12] is one of the most widely used databases for phishing identification purposes. It is a community-driven project, where the users are able to verify, track and submit details of phishing websites. Thus, this database is regularly updated since it allows the community to contribute to its dataset, making it dynamic and growing continuously.

2) DMOZ

DMOZ, also known as the *Open Directory Project* [25], is used for gathering benign URLs by providing a wide range of legitimate sites and thereby extensively used in multiple research studies involving malicious and benign URLs. This dataset is freely available and can be accessed through Kaggle.

3) UCI MACHINE LEARNING REPOSITORY

The UCI Machine Learning Repository [47] offers different datasets, which can be effectively employed in the research studies of malicious URL detection. For example, this repository contains the Phishing Web Site Dataset [48], which consists of 11055 entries of websites including both the phishing as well as benign ones, with 30 different features for each of the website entry.

4) ALEXA WHITELISTS

Alexa Whitelists dataset [49] provides a list of the most popular and frequently visited websites, which are ranked based on the network traffic. It is often used in malicious URL detection as a source of legitimate URLs. Since popular sites are less likely to be malicious, they serve as a reliable benchmark in classifying URLs as benign webpages.

5) OpenPhish

OpenPhish [15] dataset is an another database commonly used for phishing URLs. It provides both the free and paid access to phishing URLs. OpenPhish provides timely and precise updates in its database and thereby helps the researchers for testing the accuracy of their models with the up-to-date malicious URL datasets.

6) URLhaus DATASET

URLhaus dataset [13] provides plain-text malware URL lists, which are generated for every 5 minutes. It aims to share the details of prominent malicious URLs that are being manipulated to spread malware, so that users can be protected from phishing attacks.

7) WEBSPAM DATASET

WEBSPAM dataset [26] includes versions of publicly available datasets such as webspam-UK2006 and webspam-UK2007 datasets. This site provides details of both the spam and non-spam URLs, which can be manually verified and can be utilized in URL-based research studies.

REFERENCES

[1] *Cyber Pravaha'-Indian Cybercrime Coordination Centre (I4C) Newsletter*. Accessed: Jan. 23, 2025. [Online]. Available: https://i4c.mha.gov.in/cyber-pravaha.aspx

[2] M. D. Karajgar, S. Sawardekar, S. Khamankar, N. Tiwari, M. Patil, V. K. Borate, Y. K. Mali, and A. Chaudhari, "Comparison of machine learning models for identifying malicious URLs," in *Proc. IEEE Int. Conf. Inf. Technol., Electron. Intell. Commun. Syst. (ICITEICS)*, Jun. 2024, pp. 1–5.

[3] E. Blancaflor, "A comprehensive review of neural-network based approaches for predicting Phishing websites and URLs," in *Proc. 5th Int. Conf. Indus. Engg. (AI)*, 2024, pp. 1–6.

[4] Y. Liang, Q. Wang, K. Xiong, X. Zheng, Z. Yu, and D. Zeng, "Robust detection of malicious URLs with self-paced wide & deep learning," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 2, pp. 717–730, Mar. 2022.

[5] S. Tyagi, S. Pingulkar, and A. Shaikh, "Comparative evaluation of machine learning models for malicious URL detection," in *Proc. IEEE Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2023, pp. 1–7.

[6] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments," *Energy Rep.*, vol. 7, pp. 8176–8186, Nov. 2021.

[7] M. Aljabri, H. S. Altamimi, S. A. Albelali, M. Al-Harbi, H. T. Alhuraib, N. K. Alotaibi, A. A. Alahmadi, F. Alhaidari, R. M. A. Mohammad, and K. Salah, "Detecting malicious URLs using machine learning techniques: Review and research directions," *IEEE Access*, vol. 10, pp. 121395–121417, 2022.

[8] *PRISMA Model*. Accessed: Dec. 2, 2024. [Online]. Available: https://www.prisma-statement.org/

[9] T. Li, G. Kou, and Y. Peng, "Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods," *Inf. Syst.*, vol. 91, Jul. 2020, Art. no. 101494.

[10] T. Tabassum, M. M. Alam, M. S. Ejaz, and M. K. Hasan, "A review on malicious URLs detection using machine learning methods," *J. Eng. Res. Rep.*, vol. 25, no. 12, pp. 76–88, Dec. 2023.

[11] J. Hong, "Phishing URL detection with lexical features and blacklisted domains," in *Adaptive Autonomous Secure Cyber Systems*. Cham, Switzerland: Springer, 2020, p. 253, doi: 10.1007/978-3-030-33432-1_12.

[12] *PhishTank-Join the Fight Against Phishing*. Accessed: Jan. 1, 2022. [Online]. Available: https://www.phishtank.com

[13] *URLhaus Dataset*. Accessed: Nov. 12, 2024. [Online]. Available: https://abuse.ch/

[14] *Google Safe Browsing*. Accessed: Dec. 23, 2025. [Online]. Available: https://safebrowsing.google.com/

[15] *OpenPhish*. Accessed: Jan. 12, 2022. [Online]. Available: https://openphish.com

[16] OpenDNS. *Company History*. Accessed: Jul. 8, 2022. [Online]. Available: https://www.opendns.com/about/company-history/

[17] H. A. Tariq, W. Yang, I. Hameed, B. Ahmed, and R. U. Khan, "Using black-list and white-list technique to detect malicious URLs," *Int. J. Innov. Res. J. Inf. Secur.*, vol. 4, pp. 1–7, Jan. 2017.

[18] R. Almeida and C. Westphall, "Heuristic phishing detection and URL checking methodology based on scraping and web crawling," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Nov. 2020, pp. 1–6, doi: 0.1109/ISI49825.2020.9280549.

[19] A. S. Rafsanjani, N. B. Kamaruddin, M. Behjati, S. Aslam, A. Sarfaraz, and A. Amphawan, "Enhancing malicious URL detection: A novel framework leveraging priority coefficient and feature evaluation," *IEEE Access*, vol. 12, pp. 85001–85026, 2024.

[20] L. A. T. Nguyen, B. L. To, H. K. Nguyen, and M. H. Nguyen, "Detecting phishing web sites: A heuristic URL-based approach," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Oct. 2013, pp. 597–602, doi: 10.1109/ATC.2013.6698185.

[21] C. Seifert, I. Welch, and P. Komisarczuk, "Identification of malicious web pages with static heuristics," in *Proc. Australas. Telecommun. Netw. Appl. Conf.*, Dec. 2008, pp. 91–96, doi: 10.1109/atnac.2008.4783302.

[22] *Alexa-Top Sites*. Accessed: Jan. 12, 2022. [Online]. Available: https://www.alexa.com/topsites

[23] *Alexa-Web Information Company's Website*. Accessed: Oct. 18, 2021. [Online]. Available: https://www.alexa.com/

[24] T. Manyumwa, P. F. Chapita, H. Wu, and S. Ji, "Towards fighting cybercrime: Malicious URL attack type detection using multiclass classification," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 1813–1822, doi: 10.1109/BigData50022.2020.9378029.

[25] *DMOZ-OpenDirectory*. Accessed: Jan. 30, 2022. [Online]. Available: https://opendirectory.org/

[26] (Jan. 21, 2022). *WEBSPAM-GitHub-Keras-Team/Keras: Deep Learning for Humans*. [Online]. Available: https://github.com/kerasteam/keras

[27] 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.

[28] F. Vanhoenshoven, G. Nápoles, R. Falcon, K. Vanhoof, and M. Köppen, "Detecting malicious URLs using machine learning techniques," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–8, doi: 10.1109/SSCI.2016.7850079.

[29] V. Vyas, A. Nair, and A. Lopes, "Heuristic based malicious URL detection," *Int. J. Res. Eng. Appl. Manage.*, vol. 6, no. 1, pp. 1–5, 2020.

[30] Y. Wang, "Malicious URL detection an evaluation of feature extraction and machine learning algorithm," *Highlights Sci., Eng. Technol.*, vol. 23, pp. 117–123, Dec. 2022, doi: 10.54097/hset.v23i.3209.

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

[32] M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee, "An adaptive machine learning based approach for phishing detection using hybrid features," in *Proc. 5th Int. Conf. Web Res. (ICWR)*, Apr. 2019, pp. 281–286, doi: 10.1109/ICWR.2019.8765265.

[33] S. N. Bannur, L. K. Saul, and S. Savage, "Judging a site by its content: Learning the textual, structural, and visual features of malicious web pages," in *Proc. 4th ACM Workshop Secur. Artif. Intell.*, Oct. 2011, pp. 1–10, doi: 10.1145/2046684.2046686.

[34] 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.

[35] 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.

[36] C. D. Xuan, H. Dinh, and T. Victor, "Malicious URL detection based on machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 1, pp. 148–153, 2020, doi: 10.14569/ijacsa.2020.0110119.

[37] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 5, pp. 2015–2028, May 2019, doi: 10.1007/s12652-018-0798-z.

[38] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," *Telecommun. Syst.*, vol. 68, no. 4, pp. 687–700, Aug. 2018, doi: 10.1007/s11235-017-0414-0.

[39] P. Nanaware, "Malicious URL detection using machine learning," *Int. J. Integr. Sci. Technol.*, vol. 2, no. 1, pp. 29–36, Jan. 2024, doi: 10.59890/ijist.v2i1.1289.

[40] O. Theobald, *Machine Learning For Absolute Beginners: A Plain English Introduction*, 2nd ed., Scatter Plot Press, 2017.

[41] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[42] Y.-T. Hou, Y. Chang, T. Chen, C.-S. Laih, and C.-M. Chen, "Malicious web content detection by machine learning," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 55–60, Jan. 2010, doi: 10.1016/j.eswa.2009.05.023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741740900445X

[43] 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.

[44] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, "URLNet: Learning a URL representation with deep learning for malicious URL detection," *ACM Conf.*, Washington, DC, USA, Jul. 2017, pp. 1–13. [Online]. Available: https://arxiv.org/pdf/1802.03162

[45] B. Yu, F. Tang, D. Ergu, R. Zeng, B. Ma, and F. Liu, "Efficient classification of malicious URLs: M-BERT—A modified BERT variant for enhanced semantic understanding," *IEEE Access*, vol. 12, pp. 13453–13468, 2024.

[46] N. A. ALfouzan and C. Narmatha, "A systematic approach for malware URL recognition," in *Proc. 2nd Int. Conf. Comput. Inf. Technol. (ICCIT)*, Jan. 2022, pp. 325–329, doi: 10.1109/ICCIT52419.2022.9711614.

[47] *UC Irvine Machine Learning Repository*. Accessed: Jan. 15, 2025. [Online]. Available: https://archive.ics.uci.edu/

[48] *Phishing Websites*. Accessed: Jan. 25, 2025. [Online]. Available: https://archive.ics.uci.edu/dataset/327/phishing

[49] *Domain Whitelisting With Alexa and Umbrella Lists*. Accessed: Jan. 27, 2025. [Online]. Available: https://isc.sans.edu/diary/Domain+Whitelisting+With+Alexa+and+Umbrella+Lists/22270

[50] S. Parekh, D. Parikh, S. Kotak, and S. Sankhe, "A new method for detection of phishing websites: URL detection," in *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Apr. 2018, pp. 949–952, doi: 10.1109/ICICCT.2018.8473085.

[51] V. Ramanathan and H. Wechsler, "PhishGILLNET—Phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training," *EURASIP J. Inf. Secur.*, vol. 2012, no. 1, Dec. 2012, doi: 10.1186/1687-417x-2012-1.

[52] A. R. Egigogo, I. Idris, M. Olalere, O. A. Abisoye, and J. A. Ojeniyi, "Development of hybridized CNN-BiGRU framework for detection of website phishing attacks," *NIPES J. Sci. Technol. Res.*, vol. 7, no. 2, pp. 263–274, Jun. 2025, doi: 10.37933/nipes/7.2.2025.18.

[53] G. Karat, J. M. Kannimoola, N. Nair, A. Vazhayil, V. G. Sujadevi, and P. Poornachandran, "CNN-LSTM hybrid model for enhanced malware analysis and detection," *Proc. Comput. Sci.*, vol. 233, pp. 492–503, Jan. 2024, doi: 10.1016/j.procs.2024.03.239.

[54] S. Baskota, "Phishing URL detection using bi-LSTM," 2025, *arXiv:2504.21049*.

[55] M. Nanda, M. Saraswat, and P. K. Sharma, "Enhancing cybersecurity: A review and comparative analysis of convolutional neural network approaches for detecting URL-based phishing attacks," *e-Prime Adv. Electr. Eng., Electron. Energy*, vol. 8, Jun. 2024, Art. no. 100533, doi: 10.1016/j.prime.2024.100533.

[56] M. Tang, M. Ye, W. Chen, and D. Zhou, "BiLSTM4DPS: An attention-based BiLSTM approach for detecting phishing scams in Ethereum," *Expert Syst. Appl.*, vol. 256, Dec. 2024, Art. no. 124941, doi: 10.1016/j.eswa.2024.124941.

[57] C. Wei, Z. Quan, Z. Qian, H. Pang, Y. Su, and L. Wang, "An attention mechanism augmented CNN-GRU method integrating optimized variational mode decomposition and frequency feature classification for complex signal forecasting," *Expert Syst. Appl.*, vol. 269, Apr. 2025, Art. no. 126464, doi: 10.1016/j.eswa.2025.126464.

[58] M. Bahaghighat, M. Ghasemi, and F. Ozen, "A high-accuracy phishing website detection method based on machine learning," *J. Inf. Secur. Appl.*, vol. 77, Sep. 2023, Art. no. 103553, doi: 10.1016/j.jisa.2023.103553.

[59] S. Al-Ahmadi, A. Alotaibi, and O. Alsaleh, "PDGAN: Phishing detection with generative adversarial networks," *IEEE Access*, vol. 10, pp. 42459–42468, 2022, doi: 10.1109/ACCESS.2022.3168235.

[60] S. A. Kamran, S. Sengupta, and A. Tavakkoli, "Semi-supervised conditional GAN for simultaneous generation and detection of phishing URLs: A game theoretic perspective," 2021, *arXiv:2108.01852*.

[61] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002. [Online]. Available: https://www.jair.org/index.php/jair/article/view/10302

[62] P. B. V. R. Rao, K. S. Pokkuluri, M. Prasad, N. Sharma, B. N. Murthy, and A. Karunasri, "Ensemble fusion for enhanced malicious URL detection by integrating machine learning and deep learning techniques," in *Proc. Int. Conf. Comput., Commun. Comput. Sci.* Cham, Switzerland: Springer, 2025, pp. 339–349. [Online]. Available: https://www.springerprofessional.de/en/ensemble-fusion-for-enhanced-malicious-url-detection-by-integrat/50616280

[63] M.-Y. Su and K.-L. Su, "BERT-based approaches to identifying malicious URLs," *Sensors*, vol. 23, no. 20, p. 8499, Oct. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/20/8499

[64] T. Bilot, G. Geis, and B. Hammi, "PhishGNN: A phishing website detection framework using graph neural networks," in *Proc. 19th Int. Conf. Secur. Cryptogr.*, 2022, pp. 428–435.

[65] C. Wang, Z. Liu, and Y. Zeng, "Detecting phishing URLs with GNN-based network inference," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Aug. 2024, pp. 504–509, doi: 10.1109/nana63151.2024.00089.

[66] S. Asiri, Y. Xiao, and T. Li, "PhishTransformer: A novel approach to detect phishing attacks using URL collection and transformer," *Electronics*, vol. 13, no. 1, p. 30, Dec. 2023, doi: 10.3390/electronics13010030.

[67] N. Q. Do, A. Selamat, H. Fujita, and O. Krejcar, "An integrated model based on deep learning classifiers and pre-trained transformer for phishing URL detection," *Future Gener. Comput. Syst.*, vol. 161, pp. 269–285, Dec. 2024.

[68] M. K. Prabakaran, P. M. Sundaram, and A. D. Chandrasekar, "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders," *IET Inf. Secur.*, vol. 17, no. 3, pp. 423–440, May 2023, doi: 10.1049/ise2.12106.

SAURABH KAILAS is currently pursuing the B.Tech degree in computer science and engineering with Manipal Institute of Technology. He has been actively working on the detection and impact analysis of malicious URLs in cybersecurity for the past 18 months. His research interests include cybersecurity, machine learning, and secure systems. He has also conducted research in computer vision, with a focus on cybersecurity applications, particularly involving the masking and de-identification of sensitive data in scanned documents.

R. ROOPALAKSHMI received the M.Tech. degree in CSE from PESIT, Bengaluru, and the Ph.D. degree in information technology from the National Institute of Technology Karnataka (NITK), Surathkal. She is currently an Associate Professor with the School of Computer Engineering, Manipal Institute of Technology (MIT), Manipal Academy of Higher Education, Manipal. She has published more than 35 research articles and served as the Principal Investigator for two funded research projects. Her research interests include image/video processing, machine learning, cybersecurity, medical image analysis, and bioinformatics.