machine learning & knowledge extraction

MDPI

Review

A Survey of Machine Learning-Based Solutions for Phishing Website Detection

Lizhen Tang * and Qusay H. Mahmoud

Department of Electrical, Computer, and Software Engineering, Ontario Tech University, Oshawa, ON L1G 0C5, Canada; qusay.mahmoud@ontariotechu.ca
* Correspondence: lizhen.tang@ontariotechu.net

Abstract: With the development of the Internet, network security has aroused people's attention. It can be said that a secure network environment is a basis for the rapid and sound development of the Internet. Phishing is an essential class of cybercriminals which is a malicious act of tricking users into clicking on phishing links, stealing user information, and ultimately using user data to fake logging in with related accounts to steal funds. Network security is an iterative issue of attack and defense. The methods of phishing and the technology of phishing detection are constantly being updated. Traditional methods for identifying phishing links rely on blacklists and whitelists, but this cannot identify new phishing links. Therefore, we need to solve how to predict whether a newly emerging link is a phishing website and improve the accuracy of the prediction. With the maturity of machine learning technology, prediction has become a vital ability. This paper offers a state-of-the-art survey on methods for phishing website detection. It starts with the life cycle of phishing, introduces common anti-phishing methods, mainly focuses on the method of identifying phishing links, and has an in-depth understanding of machine learning-based solutions, including data collection, feature extraction, modeling, and evaluation performance. This paper provides a detailed comparison of various solutions for phishing website detection.

Keywords: cybersecurity; cybercrime; phishing; phishing website detection; anti-phishing; machine learning

Citation: Tang, L.; Mahmoud, Q.H. A Survey of Machine Learning-Based Solutions for Phishing Website Detection. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 672–694. https://doi.org/10.3390/make3030034

Academic Editor: Francesco Buccafurri

Received: 7 July 2021
Accepted: 17 August 2021
Published: 20 August 2021

1. Introduction

The Internet has become an indispensable part of people's lives. It is impossible to imagine the world without the Internet. The January 2021 global digital population report shows that there are 4.66 billion active Internet users worldwide, accounting for 59.5% of the global population. Among them, 92.6% of users use smartphones to connect to the Internet [1]. The Internet has completely changed the way people live and work, such as information communication, shopping, chatting, and office work. Due to the pandemic that started at the end of 2019, many traditional industries have shifted from offline services to online services, such as catering and retail. Netizens have left a lot of sensitive data on the Internet, such as usernames, account names, passwords, privacy questions, personal information, and credit card information. Cybercriminals obtain this information through various illegal means and forge these users to carry out illegal activities on the Internet. In the early days of the invention of the Internet, network security issues have already appeared. With the development of the Internet, network attack techniques have also changed rapidly, which has brought many challenges to network security. According to the methods and forms of network attacks, cybersecurity issues are mainly divided into the denial-of-service attack (DoS), man-in-the-middle (MitM), SQL injection, zero-day exploit, DNS tunneling, phishing, and malware categories.

Phishing is a network attack that combines social engineering and computer technology to steal the sensitive personal information of users. Attackers solicit individuals to click phishing links by sending them emails, SMS, or social media messages with deceptive

content. Phishing has been around for more than 30 years, and a large number of users are deceived every year, causing economic losses. In particular, in 2020, the number of phishing attacks increased tremendously [2]. Since the COVID-19 pandemic, government departments in many countries have introduced financial assistance programs. Cybercriminals use phishing to obtain sensitive personal information, thereby fraudulently applying for government subsidies such as unemployment benefits. Among the cyber-attack complaints reported by the U.S. public in 2020, phishing network complaints accounted for the highest proportion [2]. In addition, the APWG phishing activity trends report for 2020 shows that the number of phishing attacks almost doubled in 2020 over the course of the year [3].

Anti-phishing strategies involve educating netizens and technical defense. In this paper, we mainly review the technical defense methodologies proposed in recent years. Identifying the phishing website is an efficient method in the whole process of deceiving user information. Many academic research and commercial products were published for detecting phishing websites. The traditional methods are list-based solutions that collect valid, legitimate websites to a whitelist or verified phishing websites to a blacklist and widely share the list to avoid other users being attacked. These approaches effectively prevent the reuse of the same phishing website URL, reducing the number of affected users and losses. It is widely used in real-time defensive actions since the computational time cost is very low in a single-string match algorithm. However, these methods have a significant disadvantage: the inability to detect new phishing URLs. Therefore, some innocent users will be attacked before the link is added to a blacklist. Some researchers proposed rule-based methods to recognize new fake websites. This method involved security expert experience and website analysis of phishing sites. According to the W3C standard, a basic URL consists of the protocol, subdomain, domain name, port, path, query, parameters, and fragment. Primely, rules are generated from the components of URLs, such as if the domain name is similar to other legitimate domains. In these rules, some need to request third-party services to obtain information, such as what is the registration date of the domain. When the rules are published in some technical articles, phishers learned them and then figured out new phishing URLs which did not match the rules. Afterward, cybersecurity specialists developed more rules, some based on the source codes of web pages.

Along with the development of machine learning techniques, various machine learning-based methodologies have emerged for recognizing phishing websites to increase the performance of predictions. Phishing detection is a supervised classification approach that uses labeled datasets to fit models to classify data. There are various algorithms for supervised learning processes, such as naïve Nayes, neural networks, linear regression, logistic regression, decision tree, support vector machine, K-nearest neighbor, and random forest. A practical product needs a robust solution that generally should satisfy two requirements. The first is a high accuracy and low false warning rate. Improving the model's performance requires a substantial dataset, especially for neural networks with complex structures. In addition, computational time is a crucial factor for real-time detection systems.

The primary purpose of this paper is to survey effective methods to prevent phishing attacks in a real-time environment. It presents the basic life cycle of a phishing attack as the entry point, focusing on the phase when a user clicks on a phishing link and using technical methods to identify the phishing link and alert the user. In addition to the commonly used blacklist matching and recognition methods, this paper provides an in-depth explanation of the machine learning-based URL detection technology. This paper presents the state-of-the-art solutions, compares and analyzes the challenges and limitations of each solution, and provides ideas for research directions and future solutions. The main contributions of this paper are the following:

1. A phishing life cycle to clearly capture the phishing problem;
2. A survey of major datasets and data sources for phishing detection websites;
3. A state-of-the-art survey of machine learning-based solutions for detecting phishing websites.

The rest of the paper is organized as follows. Section 2 reviews the background and related work of phishing. Section 3 lists the methodologies of detecting website phishing in terms of list-based methods, heuristic strategies, and machine learning-based solutions. In particular, the general architecture of the phishing network detecting solution based on machine learning is explained in detail. Section 4 introduces several frameworks of website phishing detection systems. Section 5 presents the state-of-the-art machine learning-based solutions, which are classified into three categories based on the number and characteristics of the learning model. Section 6 discusses the challenges of detecting phishing attacks. Section 7 shares the conclusions of this study.

2. Background and Related Work

Phishing is a common cyberattack performed by sending an email or a message to deceive recipients visiting a bogus page and then collecting users' sensitive data, such as usernames, passwords, and credit card numbers, for financial gain.

Figure 1 demonstrates the phishing life cycle. First, an attacker creates a phishing website that looks very similar to a legitimate website. On the one hand, attackers used spelling mistakes, similar alphabetic characters, and other methods to forge the URL of the legitimate website, especially the domain name and network resource directory. For example, the link "https://aimazon.amz-z7acyuup9z0y16.xyz/v" (accessed on 9 May 2021) imitates https://www.amazon.com. Although the browser on the computer can see the URL address by moving the mouse to the clickable link, it is difficult for the average user to identify these URLs with the naked eye and memory as imitating legitimate URLs. On the other hand, imitation of web content is also a key point. Typically, attackers use scripts to obtain logos, web layouts, and text from genuine web pages. Form submission pages that require user input of sensitive information are most often faked by cybercriminals, such as the login page, payment page, and find password page.

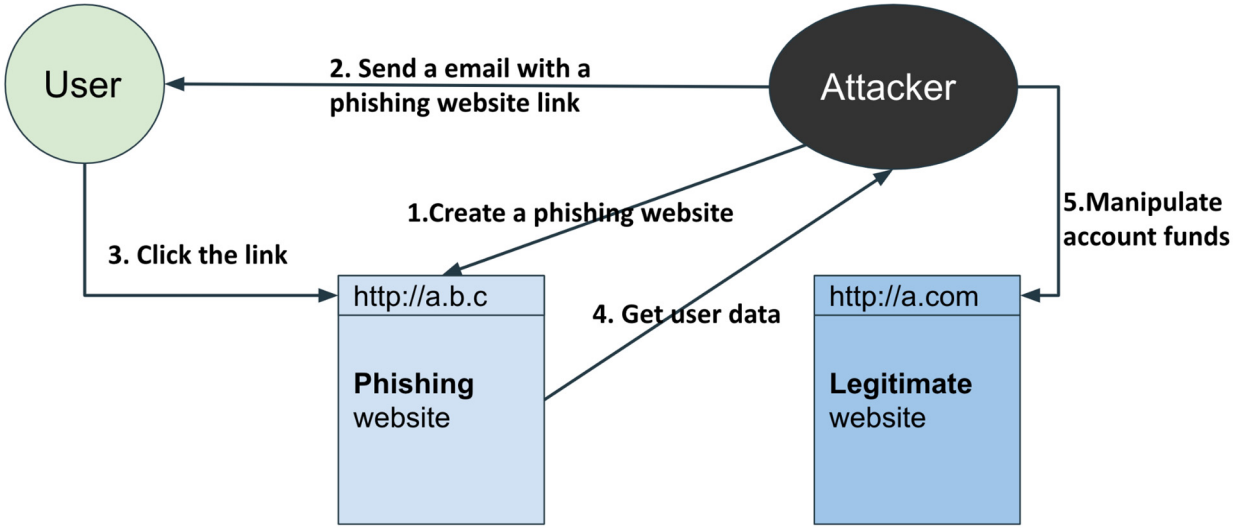

Figure 1. Phishing life cycle.

The second step is sending an email that strongly guides readers to click on the link. The way to send phishing links is not only by email but also SMS, voice message, QR codes, and spoof mobile applications [4]. With the widespread use of smartphones and social media, the number of channels for criminals to spread false information has increased. In all these ways, text and pictures are usually used to trick readers into clicking on a link. For example, the attacker imitated the customer service of a telecommunications company to send an email to urge users to pay to prevent downtime in arrears. Although scam emails are sent randomly, there is always a small number of users with weak defensive awareness who will be deceived. In this step, the attacker applied social engineering meth-

ods, including psychological manipulation, to trick users into making security mistakes. Perpetrators are good at building a sense of fear and urgency and gaining the user's trust via text messages. Afterward, the user clicks the link that will direct them to open a fake website. Particularly, real URL strings are hidden before redirecting to web browsers on mobile phones.

The next step is collecting personal information on the fake website, which looks like the real company or organization's web page, by using a similar logo, name, user interface design, and content, commonly occurring with login, reset password, payment, and renewal personal information. When users submit sensitive information to web servers that attackers build, criminals will receive all the data.

The last step is stealing the user's account funds by using the user's real information to fake the user's request for a real website. Even some individuals are using the same usernames and passwords for multiple websites. In this way, the attacker steals multiple accounts from the user. Some phishers use stolen data for other criminal activities. Since the phishing technique was recorded in a paper in 1987 [5], phishing methods have changed with the development of the Internet. For example, when online payment becomes popular, attackers target online payment phishing. According to the 2020 Internet Crime Report, the Internet Crime Complaint Center (IC3) received 791,790 cyberattack complaints, of which phishing scams accounted for approximately 30%, becoming the most complained about type of cybercrime and causing more than USD 54 million in losses [2]. Therefore, to individuals who surf on the Internet, distinguishing between real and fake web pages is vital. Users need visual tools to help users identify phishing websites.

2.1. Anti-Phishing

As we can see from Figure 1, there are five steps before an attacker steals the money from the user's account or uses the information for other attacks. Therefore, blocking any step could stop a phishing attack. Here, we discuss the method of anti-phishing starting from each step.

2.1.1. Web Scraping

Although it is hard to prevent perpetrators from creating web pages, some techniques could increase their costs. Attackers will use scripts to write crawlers to obtain legal web pages' content automatically and then intercept useful information and copy it to phishing web pages. Therefore, legitimate websites could prevent web scraping by several techniques in respect to obfuscation using CSS sprites to display important data, replacing text with images.

2.1.2. Spam Filter

Spam filtering techniques are used to identify unsolicited emails before the user reads or clicks the link. Some mainstream email services have integrated spam filtering components, such as Gmail, Yahoo, Outlook, and AOL. The initial filters relied on blacklists or whitelists and empirical rules. With the development of artificial intelligence technology, some filters also integrate intelligent prediction models based on machine learning to filter out spam that is not on the list. For example, Gmail could block approximately 100 million extra spam emails daily with the machine learning-based spam filter [6].

2.1.3. Detecting Fake Websites

When users visit a phishing web page that looks like a legitimate website, many people do not remember the legitimate website's domain name, particularly for some start-ups or unknown companies, so users cannot recognize the phishing website based on the URL. Some web browsers integrate a security component to detect phishing or malware sites, such as Chrome, which will display warning messages when one visits an unsafe web page. Google launched Google Safe Browsing in 2007, and it is integrated into many Google products, such as Gmail and Google Search. Google Safe Browsing is a security component

based on a blacklist that contains malware or phishing URLs [7]. In addition, there are several web browser extensions for detecting phishing websites. However, the blacklist or whitelist-based solutions are invalid for unknown phishing websites. Fortunately, the rapid development of artificial intelligence technology has brought new ideas and solutions to detecting phishing attacks. The predictive model based on machine learning can identify phishing links that are not on the whitelist and circumvent existing rules.

2.1.4. Second Authorization Verification

After the attacker obtains the user's sensitive data, the next step is to use the data to log in to the legitimate website, operate the account, and steal funds. Therefore, when the website detects that the IP address and device information of the user who is logging in does not match the commonly used information, it becomes crucial to add steps to verify the authenticity of the user. Usually, the extra verifications are dynamic and biological, such as facial movement, expression recognition, or voiceprint recognition.

2.2. Related Work

Many survey papers have been published introducing and comparing different solutions for detecting phishing websites. Basit et al. reported a survey on artificial intelligence-based phishing detection techniques. The authors analyzed the harm and trends of phishing attacks from statistical phishing reports [8]. They collected major communication media and target devices during phishing attacks and listed various attack techniques. The paper focuses on anti-phishing measurements, which are classified into four sections: machine learning, deep learning, hybrid learning, and scenario-based. Each section presents several major algorithms and conducts a comparison among those algorithms. In addition, they draw several conclusions by reading various state-of-the-art solutions, stating that machine learning-based solutions are widely used and effective, the feature selection process contributes high-grade performance, high accuracy often requires more computing resources, and the random forest model obtains the highest accuracy.

Singh et al. conducted a review on machine learning-based phishing detection [9]. The authors introduced a brief history of phishing and major phishing attack reports. In the paper, phishing attacks are divided into two types: social engineering attacks and malware-based phishing. They classified features into three categories—source code features, URL features, and image features—which are all based on rules.

In 2020, Vijayalakshmi et al. presented a survey on major detection techniques and taxonomy for detecting phishing [10]. A statistical report from APWG shows the trend of phishing attacks from 2017 to 2019. In the paper, a taxonomy of automated phishing detection solutions was introduced, which classified all the solutions into three categories depending on the input parameters: web address-based methods, webpage content-based solutions, and hybrid approaches. According to the techniques applied in the solutions, web address-based approaches were divided into list-based, heuristic rule-based, and learning-based approaches, and web content-based solutions were separated into rule-based and machine learning-based solutions. The authors listed most of the state-of-the-art methodologies for each category and interpreted the details of every solution. After comparing all methods by several evaluation metrics, such as classification performance, limitations, third-party service independence, and zero-hour attack detection, they suggested that hybrid approaches would obtain a high accuracy rate and be suitable for real-time systems and that deep learning-based solutions will be a valuable direction in the future.

Kalaharsha and Mehtre surveyed phishing detection solutions that were classified into several categories based on the techniques and input parameters applied. In the paper, different types of phishing attacks and three phishing techniques are introduced [11]. The authors listed 18 methods and 9 datasets for detecting phishing websites and compared the accuracy performance among all the models. In addition, some challenges are presented in the paper, such as reducing false-positive rates and overfitting.

More recently, Jain and Gupta presented a comprehensive survey on analyzing phishing attack techniques, detection methods, and some existing challenges [12]. They imported statistical reports and motivation of phishing attacks and presented different phishing attack techniques on PCs and smartphones. Then, the authors introduced various defense methods and compared existing anti-phishing approaches published from 2006 to 2017 for their advantages and limitations. Afterward, several major challenges were presented, such as selecting efficient features, identifying tiny URLs, and detecting smartphones.

3. Methodologies of Phishing Website Detection

Since phishing is a social engineering issue, effective countermeasures are built for different aspects in terms of education, legal supervision, and technical approaches [4]. This survey focuses on technical strategies for detecting phishing websites. The methodologies of detecting phishing websites are developed, which are divided into three categories: list-based, heuristic, and machine learning methods [13]. The list-based approaches consist of whitelists and blacklists that have been manually reported and confirmed by systems. A whitelist is a set of validated legitimate URLs or domains. Obviously, a blacklist is a group of approved phishing websites. Since one user reported and verified the website as a phishing website, the URL will be added to the blacklists, which could be used to prevent other users from being disrupted. Heuristic strategies identify a phishing web page depending on a group of features extracted from the textual contents of the website and compare the features with the legitimate one. The idea of the approach is that the attackers usually deceive users by imitating well-known websites. The machine learning methods also depend on the features from the website, build the model to learn from a batch of data with structured features, and then predict if the new website is a phishing website. In the machine learning area, detecting phishing websites is a classification problem.

3.1. List-Based Approaches

Jain and Gupta proposed an auto-updated, whitelist-based approach to protect against phishing attacks on the client side in 2016. The experimental results demonstrate that it achieved 86.02% accuracy and less than a 1.48% false-positive rate, which indicates a false warning for phishing attacks. The other benefit of this approach is fast access time, which guarantees a real-time environment and products [14].

3.2. Heuristic Strategies

Tan et al. introduced a phishing detection approach named PhishWHO, which consists of three phases. First, it obtains identity keywords by a weighted URL token system and ensembles the N-gram model from the page's HTML. Secondly, it puts the keywords into mainstream search engines to find the legitimate website and the legal domain. Next, it compares the legal domain and the target website's domain to determine if the target website is a phishing website or not [15]. Chiew et al. used a logo image from the website to distinguish if the website was legal [16]. In this paper, the authors extracted a logo from web page images by some machine learning algorithms and then queried the domain via the Google search engine with a logo as a keyword. Therefore, some researchers also called this category search engine-based approach.

3.3. Machine Learning-Based Methods

Machine learning-based countermeasures are proposed to address dynamic phishing attacks with higher accuracy performance and lower false positive rates than other methods [4]. Consequently, the machine learning approach consists of six components: data collection, feature extraction, model training, model testing, and predicting. Figure 2 shows the flowchart of each part. Existing machine learning-based phishing website detection solutions are based on this flowchart to optimize one or more parts to obtain better performance.

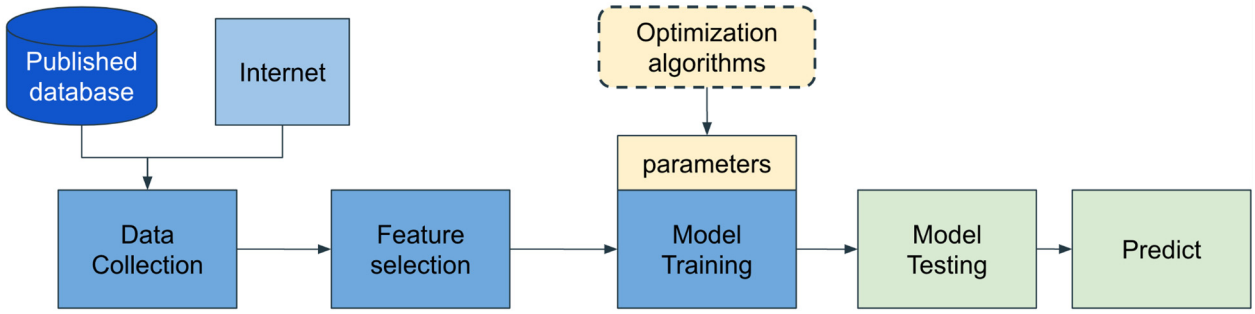

Figure 2. Machine learning flowchart for detecting phishing websites.

3.3.1. Data Collection and Feature Extraction

Data are the source of each approach and proves to be a vital influence for the performance. There are two methods to collect data: loading published datasets and pulling URLs directly from the Internet. Table 1 shows several major data sources. In these three published datasets, every row's data object contains several features extracted from a URL and a label of classes. The original URL strings could be collected from websites by running open API or data mining scripts.

Mohammad et al. proposed an automatic technique to extract phishing website features and weigh the importance of each feature in 2012 [17]. In that paper, the authors collected 2500 phishing URLs from the phishTank archive [18] and extracted 17 features which were classified into three categories: address bar-based features, abnormal-based features, and HTML and JavaScript-based features. Most of the features were automatically extracted from the URL and the source code of the web page without relying on third-party services. However, the age of the domain and DNS record were extracted from the WHOIS database [19]. The rank of the web page was obtained from the Alexa database [20]. Meanwhile, the authors described an IF-ELSE rule and set a weight for each feature. The weight of a feature came from the calculation of the feature value for phishing accounts for the total number of phishing links. Each feature's value could be numeric as an element of the set $\{1, 0, -1\}$, respectively, each standing for legitimate, suspicious, and phishing in turn [21].

In 2015, Mohammad et al. published a phishing website dataset on the UCI Machine Learning Repository, which became a foundation for machine learning-based phishing detection solutions and was widely used in many related research areas, containing 11,055 instances with 30 features [22]. Furthermore, Choon published a phishing dataset on Mendeley in 2018, containing 10,000 data rows with 48 features extracted from phishTank and OpenPhish for phishing webpages and Alexa and Common Crawl for legitimate webpages, each having 5000 websites [23].

As we can see, the published datasets are small datasets compared with other machine learning programs. Therefore, some resampling techniques are involved in the process, such as N-fold cross-validation, which splits the data into N pieces iterated N times, with each iteration selecting one piece data as testing data and others as training data. On the other hand, some researchers collected URLs from the Internet, such as from phishTank, OpenPhish, and Spamhaus.org for phishing URLs and dmoztools.net, Alexa, and Common Crawl for legitimate websites, and then parsed the features by themselves.

With the successful development of the natural language processing (NLP) technique, many researchers capture character-level features from URL strings based on the NLP and then feed them into deep learning models to increase the accuracy. The significant advantages of this method are irrelevant cybersecurity expertise and not relying on third-party network services [24]. Since the characters in the URL are continuous, it is difficult to distinguish words and have no semantics. Character-level features are used, such as character-level TF-IDF features. TF-IDF means Term Frequency–Inverse Document Frequency. The character level stands for each character as a term. The algorithm calculates

each character's TF-IDF score and then generates a matrix with those scores, which stands for the relevance of a character in the URL string. Using "https://www.google.com/" (accessed on 18 July 2021) as an instance, it has 17 characters ("h", "t", "t", "p", "s", ":", "//", "w", "w", "w", ".", "g", "o", "o", "g", "l", "e", ".", "c", "o", "m") and is called character level 17-g in the corpus. Therefore, it will generate a vector with 17 TF-IDF scores. One character's TF-IDF score is calculated as in the math formulation shown below:

$$\mathrm{TF}(t, d) = \frac{\text{Number of times character t appears in a document d}}{\text{Total number of characters in the document d}} \quad (1)$$

$$\mathrm{IDF}(t, D) = \log_e\left(\frac{\text{Total number of documents D}}{\text{Number of documents with character t in it}}\right) \quad (2)$$

$$\mathrm{TFIDF}(t, d, D) = \mathrm{TF}(t, d) \times \mathrm{IDF}(t, D) \quad (3)$$

Table 1. Major data sources for detecting phishing websites.

Data Source	Type	Remarks
UCI [22]	Published dataset	11,055 instances with 30 features
Mendeley [23]	Published dataset	10,000 instances with 48 features
ISCX-URL-2016 [25]	Published dataset	35,000 legitimate URLs 10,000 phishing URLs
https://phishtank.com (accessed on 18 July 2021) [18]	Website	Valid phishing URLs
https://openphish.com (accessed on 18 July 2021)	Website	Valid phishing URLs
https://commoncrawl.org/ (accessed on 18 July 2021)	Website	Legitimate URLs
https://www.alexa.com/ (accessed on 18 July 2021) [20]	Website	Legitimate URLs

3.3.2. Feature Selection

Feature selection is the process of automatically selecting important features which contribute the most to the machine learning model. Having closely relevant features in the input can enhance the performance of the model, decrease training time (especially in deep learning models), and reduce overfitting issues. Generally, feature selection methodologies could be classified into three categories: the filter method, wrapper method, and embedded method.

Zamir et al. utilized recursive feature elimination (RFE), information gain (IG), and relief ranking to omit redundant features for phishing detection. Furthermore, they introduced principal components analysis (PCA) for analyzing attributes [26,27]. IG is an indicator that tells us the importance of features by calculating class probability, feature probability, and class probability under a feature condition. RFE is a widely used feature reduction algorithm to remove the least essential features in the training process until the error rate meets expectations.

A relief ranking filter is a feature value filtering algorithm that calculates the feature value score by comparing the feature values of two adjacent data points discovered by the nearest neighbor search algorithm and then sorts them to obtain the feature value weight according to the score. Shabudin et al. used this algorithm to apply to the UCI dataset for phishing website classification. After the feature selection process, they obtained 22 features with weights ranking and removed 8 redundant features of zero scores [28].

Zabihimayvan and Doran applied Fuzzy Rough Set (FRS) theory to select important features from the UCI dataset and Mendeley dataset for phishing detection applications [13].

Fuzzy Rough Set (FRS) theory is an extension of Rough Set (RS) theory. RS is a method to find a decision boundary by calculating the equality of each data point based on certain features and the same classes, such as websites A and B both being phishing websites and their features a and b having the same value. RS is suitable for the original UCI dataset in which the features are utilized as a discrete value; that is, they are an element of set $\{-1, 0, 1\}$. However, after the dataset executes the nominalization process, the value of the feature is transferred to a continuous number from 0 to 1, and the FRS strategy is applied.

El-Rashidy introduced a novel technique to select features for a web phishing detection model in 2021 [29]. The feature selection method contains two phases. The first phase calculates each feature's absence impact by training the random forest model with a new dataset that removes one feature and figures out the accuracy. After the absence of each element in the loop, a feature queue ranked from high to low accuracy is obtained. The second stage is to train and test the model by starting from one feature, adding a new feature from the ranked feature list each time to form a dataset, calculate the accuracy of each time, and finally find the feature subset with the highest accuracy. This method works to select the most effective feature subset. However, since each new dataset must go through the algorithm training and testing process, a high computational complexity and a long calculation time are involved. For example, if the UCI dataset has 30 eigenvalues, then the first stage loops 30 times, the second stage loops 30 times, and the tree algorithm training must be performed each time. Therefore, this methodology is suitable for small feature sizes and single classifiers.

3.3.3. Modeling

Machine learning-based models can be classified into three categories: single classifier, hybrid models, and deep learning. Hybrid models combine more than one algorithm applied to the training process. Phishing website detection is a binary classification problem. Some widely used classification algorithms are listed below.

SVM: A support vector machine (SVM) is a supervised learning algorithm that classifies data points into two sections and predicts new data points belonging to each section. It is suitable for linear binary classification, which has two classes labeled, and the classifier is a hyperplane with N dimensions relevant to the number of features. The core idea of this algorithm is to maximize the distance between the data point and the segmentation hyperplane. For example, there are two classes—phishing and legitimate—and a 29-dimension hyperplane when we use the UCI dataset for training the SVM model.

Decision Tree: A decision tree is a popular machine learning algorithm, and the model logic is a tree structure. Each node in the decision tree is a feature; each stem presents a feature value and a possibility, and the last node presents the result. The more straightforward tree structure tends to have better performance. When trees grow very deep, it likely leads to overfitting training datasets.

Random Forest: A random forest is an ensemble of decision trees for classification and regression. Random forests reduce the overfitting problem by classifying or averaging the output of individual trees in training processing. Therefore, random forests generally have higher accuracy than decision tree algorithms.

k-NN: A k-nearest neighbors' algorithm (k-NN) is a non-parametric classification algorithm that makes predictions by finding similar data points through calculating the distance between the target and the nearest neighbors. There are some methods to calculate the distance with respect to the Euclidean distance for continuous data and the Hamming distance for discrete values. In particular, it does not have a training process, and each prediction will take a long time. Therefore, this algorithm is generally not suitable for real-time scenarios.

Bagging: Bagging, also called bootstrap aggregating, is an ensemble meta-learning algorithm for improving other machine learning algorithms' performance in classification and regression. The bootstrapping procedure divides the original training dataset into N pieces and uses resampling techniques to generate the same size of the original dataset

in each piece and then conducts classification in N iterations that could be executed in parallelization. Finally, the aggregating process combines N classifier outputs by averaging or voting.

Naive Bayes: A naive Bayes classifier is a probabilistic statistical algorithm based on Bayes' theorem with robust independence features. Bayes' theorem is a conditional probability theory. It is also called simple Bayes and independence Bayes.

In recent years, more and more researchers have used hybrid classification in phishing website detection approaches to achieve higher performance and lower computational times than single classifiers. Most hybrid models are based on a primary learner, with the addition of an algorithm for feature selection or optimizing the initialization parameters of the basic algorithm, such as hyperparameters for neural networks.

Since the rapid development of deep learning and the success of natural language processing (NLP), researchers have proposed diverse deep learning models which derive information and sequential patterns of URL strings without depending on the source code features extracted from the web page content. It does not require professional cybersecurity knowledge of phishing and depends on third-party services to capture characteristics [24]. Some broadly used deep learning algorithms are listed below.

CNN: A convolutional neural network (CNN) is a feedforward deep learning algorithm and is widely used in image classification. The regular architecture of a CNN consists of multiple layers, followed by the input layer, hidden layers, and output layer. Commonly, the hidden layers have convolutional layers, pooling layers, and fully connected layers.

RNN: A recurrent neural network (RNN) is a deep neural network with an internal memory function to handle diverse length sequences of inputs, such as text. Therefore, it has been successfully applied in text mining.

Table 2 shows a summary of these algorithms based on the same dataset. We used the Big O notation to measure the computational complexities of machine learning algorithms. The complexity of a deep neural network depends on the architecture of the networks. Generally, it needs to compute the activation function of all neurons. Interpretability presents the difficulty of understanding how the model works. Traditional machine learning algorithms are user-friendly models. In deep neural networks, it is hard to know which neuron is playing what role and which input feature contributes to the model output. In contrast, deep neural networks require more training data than other algorithms to obtain acceptable performance. The significant advantage of deep neural networks is dealing with text data, such as URL strings.

Table 2. Machine learning algorithms for detecting phishing websites. Here, n is the number of training instances, and d is the number of dimensions of the data.

Algorithm	Training Time Complexity	Interpretability	Training Data Size	Inputs
Support Vector Machine (SVM)	$O(n^2)$	Median	Small	Structured data
k-nearest neighbors (k-NN)	O(knd) k = number of neighbors	Median	Small	Structured data
Decision Tree	O(nd log n)	High	Small	Structured data
Random Forest	O(knd log n) k = number of trees	Median	Small	Structured data
Naïve Bayes	O(nd)	High	Small	Structured data
Deep Neural Networks	Compute the activation of all neurons	Low	Large	Structured data or text data

3.3.4. Performance Evaluation

The evaluation of performance was carried out during the testing process. The original dataset would be divided into training data and test data, usually 80% and 20%, respectively. When evaluating the classifier's behavior on the testing dataset, there were four statistical numbers: the number of correctly identified positive data points (TP), the number of correctly identified negative data points (TN), the number of negative data points labeled by the classifier as positive (FP), and the number of positive data points labeled by the model as negative (FN). This is shown in Table 3.

Table 3. Four statistical numbers of predicting results.

True Labels	Labels Returned by the Classifier in the Testing Process	
	Positive	**Negative**
Positive	TP [1]	FN [2]
Negative	FP [3]	TN [4]

[1] The number of correctly identified positive data points. [2] The number of positive data points labeled by the model as negative. [3] The number of negative data points labeled by the classifier as positive. [4] The number of correctly identified negative data points.

There are several broadly used metrics to evaluate performance. The classification accuracy is the ratio of correct predictions to total predictions:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{4}$$

In binary classification cases, it is known that random selection has 50% accuracy. In unbalanced datasets, sometimes high accuracy does not mean that the model is excellent. For instance, among the 10,000 data, 9000 were legitimate websites, and 1000 were phishing websites, so when the prediction model did nothing, it could reach 90%. Accuracy is misleading when the class sizes are substantially different. Precision is the percentage of correctly identified positive data points among those predicted as positive by the model. The number of false-positive cases (FP) reflects the false warning rate. In real-time phishing detection systems, this directly affects the user experience and trustworthiness:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

The recall is the portion of positive data points labeled as such by the model among all truly positive data points. The number of false-negative cases (FN) represents the number of phishing URLs that has not been detected. Leak alarms mean that users are likely to receive an attack that could result in the theft of sensitive information. Misleading users can do more harm to users than not detecting them:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

The F-measure or F-score is the combination of precision and recall. Generally, it is formulated as shown below:

$$\text{F}_\beta = \frac{\left(\beta^2 + 1\right) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \quad \beta \in (0, \infty) \tag{7}$$

Here, β quantifies the relative importance of the precision and recall such that $\beta = 1$ stands for the precision and recall being equally important, which is also called F1. The

F-score does the best job of any single statistic, but all four work together to describe the performance of a classifier:

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{recall}}{\mathrm{Precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})} \quad (8)$$

In addition, many researchers use the N-fold cross-validation technique to measure performance for phishing detection [4,30,31]. The N-fold cross-validation technique is widely used on small datasets for evaluating machine learning models' performance. It is a resampling procedure that divides the original data samples into N pieces after shuffling the dataset randomly. One of the pieces is used in the testing process, and others are applied to the training process. Commonly, N is set as 10 or 5.

4. Frameworks of Phishing Website Detection Systems

The goal of anti-phishing research is to prevent individual Internet users from suffering phishing attacks. With the development of anti-phishing research, phishing attackers are constantly updating their technology. The naked eye does not recognize many phishing links well, and individual netizens need tools to help identify them. Due to the tools and methods of the phishing network, many researchers naturally think of expanding on the browser. The following two methods are based on the browser.

4.1. Anti-Phishing Web Browser

In 2020, HR et al. built a web browser with a phishing detection component [32]. The regular web browser had two core engines—a browser engine and a render engine—which are responsible for connecting to the Internet to fetch the web page via the URL, parsing the web page by XML, HTML, CSS, JAVASCRIPT interpreters, storing cookies, etc. The proposed browser added an intelligent engine to detect phishing websites between the browser engine and render engine. When a user input a URL, the intelligent engine started to predict if the target website was a phishing website and afterward sent the result to the render engine. If the predicted result showed a phishing website, the render engine would pop a warning message to the user interface. This paper used the random forest algorithm to train the model, and it obtained 99.36% accuracy and a 0.64% false-positive rate on the UCI dataset with 30 rule-based features.

4.2. Web Browser Extensions

Armano et al. introduced a real-time client-side phishing prevention solution [33]. The approach contains a built-in JavaScript frontend and a built-in Python backend. The frontend collects the web page source code and handles the user interface and interaction with the backend, analyzing the website and predicting if it is a phishing website. The backend consists of a disputer for checking against the whitelist, a phishing detector for predicting the website's legitimacy, and a target identifier to find the legitimate website relevant to the input URL based on the logo, keywords, and other content. The phishing detector is implemented by an existing solution that uses the gradient boosting algorithm as the classifier [34]. The authors experimented with 200 phishing websites to monitor the response time. The results showed that the response time for a phishing URL was longer than a legitimate one, which was approximately 2 s, and the appearance of the alert cost occurred in less than 500 ms.

With the rapid development of the mobile Internet, many user behaviors have shifted from the PC to the smartphone. Therefore, phishing website monitoring on mobile phones is vital.

4.3. Mobile Applications

Kadhim et al. developed a web browser application on Android smartphones to predict phishing websites based on the UCI dataset with 30 features extracted from the URL and source code [35]. The application compared different classification algorithms in

training processing, such as the decision table, J48, SVM, Bayes Net, and random forest model, which outperformed the others with 97% accuracy. The authors conducted the experiments on Samsung and Nexus phones running the Android 5.1 operating system.

In addition to the framework mentioned in academic papers, there are also several published Internet products. We list some of the more used products in Table 4.

Table 4. Released phishing detection products.

Name	Type	Devices	Techniques	Advantages	Shortcomings	Users
Phish Detector [36]	Web browser extension	Chrome	Rule-based	Zero false-negative alarms	Only for online-banking web sites	2000+
Netcraft Extension [37]	Web browser extension	Chrome	Blacklist-based	Multiple features, including coronavirus-related cybercrime.	New phishing attacks cannot be prevented	50,000+
WOT [38]	All	Browser Mobile PC	Blacklist + machine learning algorithms	Multi-platform security service	Charged	1,000,000+
Pixm Phishing Protection [39]	Web browser extension	Chrome	Deep learning algorithm	Advanced anti-phishing solution (AI)	Charged	1000+
Sharkcop [40]	Web browser extension	Chrome	SVM algorithm	New attacks can be detected Few features are used	The project is currently on hold Feature extraction relies on third-party services, such as domain age	-
PhishFort [41]	Web browser extension	Chrome Firefox	Blacklist-based	Free	New phishing attacks cannot be prevented	2000+

5. State-of-the-Art Machine Learning-Based Solutions

In recent years, massive phishing detection solutions were proposed and achieved high accuracy. It is believed that the recently proposed methods are more advanced. Several major state-of-the-art methodologies are listed below, and they are classified into three categories.

5.1. Single Classifier

In 2021, Gupta et al. developed a lightweight phishing detection approach and achieved 99.57% accuracy with the random forest algorithm [42]. The authors extracted 19,964 instances with 9 lexical features from the ISCX-URL-2016 dataset published by the University of Canada Brunswick [25]. The ISCX-URL-2016 dataset contains more than 35,300 legitimate URLs and approximately 10,000 phishing URLs taken from an active repository of phishing sites https://openphish.com (accessed on 18 July 2021). To balance the distribution of the two classes, the authors randomly filtered 10,000 benign URLs and 9964 phishing URLs. Furthermore, the Spearman correlation algorithm and K best algorithm are applied to figure out the feature importance. Based on other previous research, nine lexical features from URLs were proposed in the paper. Afterward, they cleaned the data by replacing the null and unlimited values with mean values and normalized them

by scaling the values between 0 and 1. Normalization is one of the important data preprocessing procedures to guarantee that one feature is not dominated by others. In addition, they used a one-hot encoding algorithm to transfer the labels to numerical values. Once the dataset is regularized, it is divided into a training dataset and a testing dataset with eight-to-two ratios. In the process of modeling, they compared four single classifiers with the performance and computational time. Finally, it was concluded that random forest had the highest accuracy rate and the lowest false positive rate. However, in terms of response time, SVM performed better. Figure 3 demonstrates the process of data preprocessing.

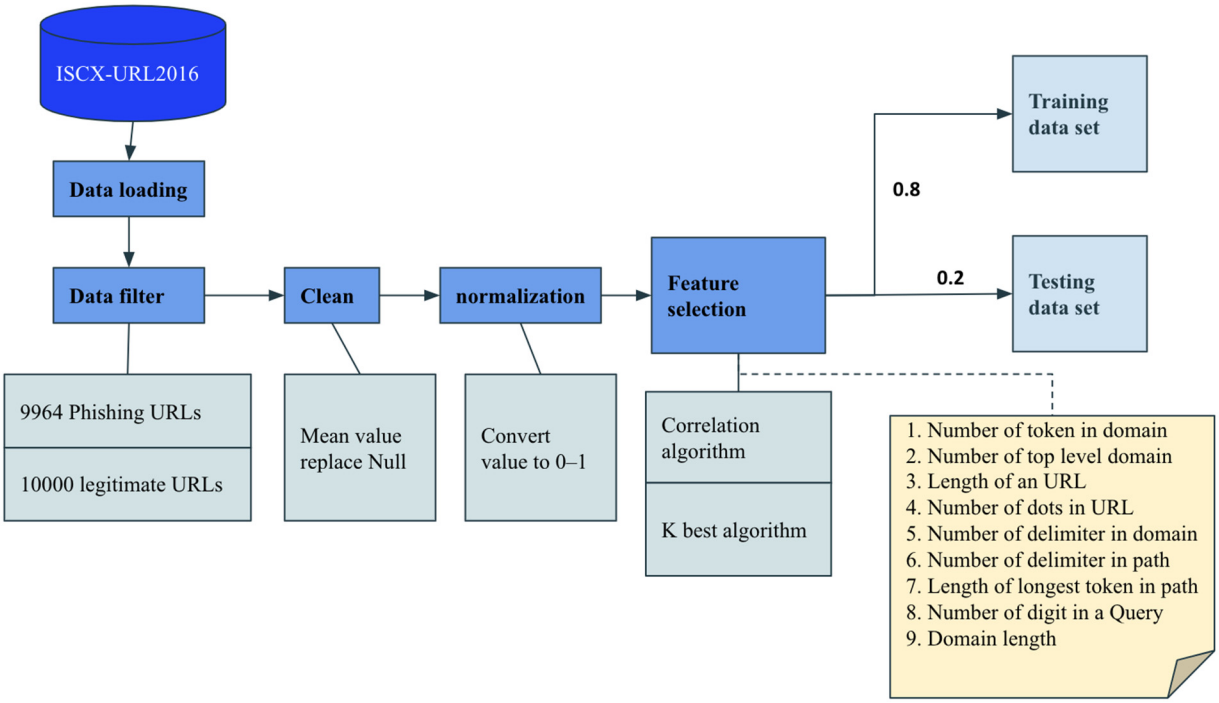

Figure 3. The data preparation process for the ISCX-URL-2016 dataset.

5.2. Hybrid Methods

In 2020, Alsariera et al. proposed four hybrid models named ABET, RoFET, BET, and LBET, each combining a meta-learner model and the extra tree algorithm, which is the basic classifier. Four meta-learner models, called Adaboost.M1, Rotation Forest, Bagging, and LogitBoost, were implemented by a meta-algorithm or metaheuristic, a high-level procedure designed to find an optimal solution for an optimization problem. This paper used 10-fold cross-validation to resample the UCI dataset and then iterated 10 times for training and testing the extra tree model, evaluated based on a weighted average value. The Adaboost.M1 model was used with a base classifier to improve performance by iterating 100 times to adjust the weights. The RoFET model used a principal component filter in the training process to achieve a high true-positive rate and decrease bias. The BET combined the bagging algorithm and extra tree algorithm executed 150 times over a resampled dataset. The LBET is a logistic regression extra tree that conquers abnormal data points, such as noise and outliers. The experimental results demonstrated that all four fusion models obtained significant performance, with over 97% accuracy, false-negative rates less than 0.038, and false-positive rates less than 0.019 [4].

Zamir et al. introduced diverse machine learning algorithms for detecting phishing websites, comparing accuracy performance from the single classifier to the stacking models. First, the authors conducted a data preprocessing procedure containing feature selection, nominalization, and principal components analysis (PCA), a dimension reduction method. The feature selection process involved a variety of algorithms in analyzing the importance of features based on the UCI dataset, such as IG, GR, Relief-F, and RFE. In comparing the

experimental results, they concluded that RFE was the efficient algorithm to eliminate unimportant features. Afterward, the features were used to fit the stacking model with a 10-fold cross-validation technique. They built two stacking models; one was combined random forest (RF), neural network (NN), and bagging (Bagging) algorithms, and the other was associated with the k-nearest neighbors, random forest, and bagging algorithms. The RF-NN-Bagging approach outperformed all other models introduced in the paper with respect to accuracy performance, which was 97.4% [26]. Figure 4 adapted from ref. [26] depicts the proposed framework.

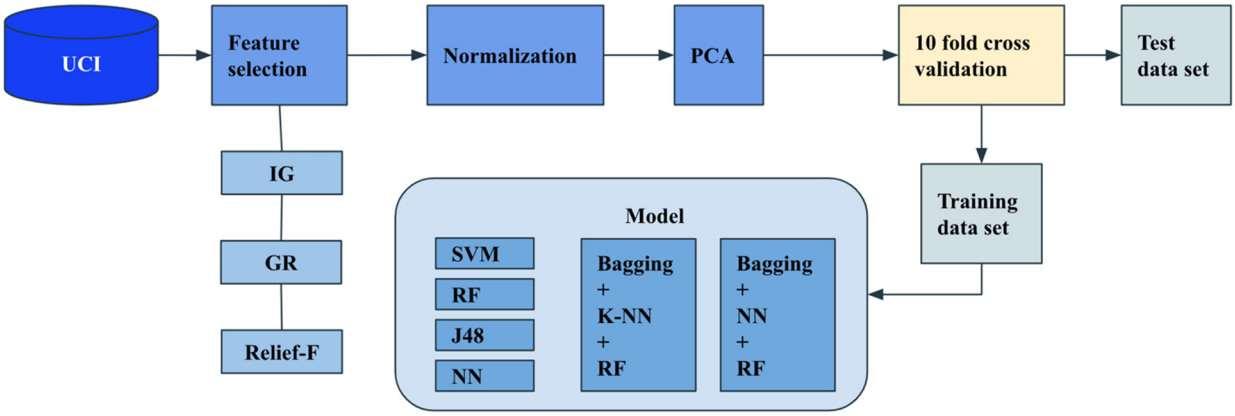

Figure 4. Hybrid algorithms for phishing detection. Information Gain (IG), Gain Ratio (GR), and Relief-F are widely used feature selection algorithms. Principal components analysis (PCA) is a reducing dimension technology by transforming data points coordinates. The UCI dataset [22] is a public phishing dataset. Support Vector Machine (SVM), Random Forest (RF), neural network (NN), and k-nearest neighbors (K-NN) are classifiers.

5.3. Deep Learning

Deep learning is a subset of machine learning which is built with deep structured architectures. There are some commonly used deep learning algorithms, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks. With the rapid development of natural language processing (NLP) and deep learning algorithms, various deep learning-based solutions are introduced for phishing detection. Figure 5, adapted from ref. [8] shows the basic architecture of deep learning-based approaches.

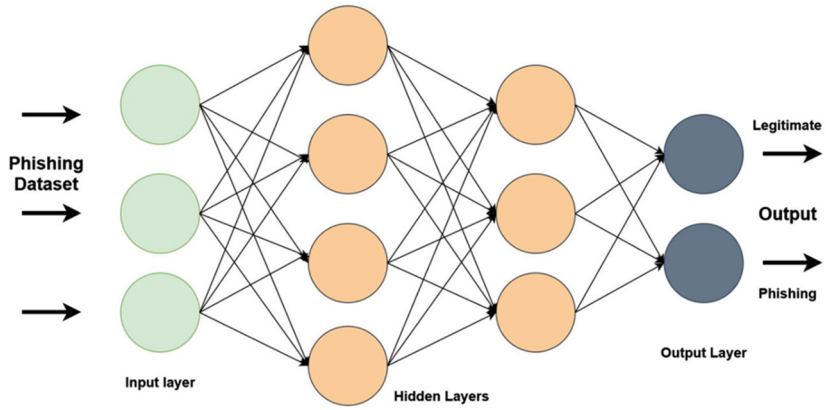

Figure 5. Deep learning for phishing detection.

Ali and Ahmed developed an intelligence phishing detection model which combined deep neural networks (DNNs) and genetic algorithms (GAs) [43]. A DNN is a well-known deep learning technique with more than two hidden layers, an input layer, and an output layer, commonly used to classify multiple labels from big data. The GAs are inspired

by the biological evolution of the genes in nature and are widely used for optimization problems that aim to minimize or maximize the value of objective functions under some restraints. In this approach, the authors regarded the problem of feature selection as an optimization problem. Mathematically speaking, the objective function minimizes the number of features, and the constraint function is the accuracy of the classification model. Meeting performance requirements with minimal features reduces the model training time and could remove the noisy data. Therefore, the GA was applied to find the optimal subset of features by computing the accuracy of the DNN model in each generation. A chromosome represents a group of features, and each gene with a binary value stands for each feature, where one is for selecting this feature and zero is not. The classification phase used the selected features as input features and the UCI dataset as a training dataset to fit the DNN model. However, the GA-DNN model got a relatively low accuracy result, which was 89%. It is known that hyperparameters and the size of a training dataset significantly affect the performance of deep learning models [8].

In 2020, Aljofey et al. proposed an efficient convolutional neural network (CNN) model for phishing detection only based on URLs [24]. They extracted character-level features from the original URLs, which were collected from different phishing websites and benign websites. The experimental results showed that this model obtained an accuracy of 95.02% on their own dataset with 318,642 instances. Wang et al. introduced a fast model called PDRCNN that used the URL string as an input, extracted features by an RNN and CNN, and then classified them with the Sigmoid function [44]. The authors collected approximately 500,000 instances from Alexa.com [20] and phishTank.com [18] and extracted semantic features based on the word embedding technique, encoding the URL string to a tensor, an input of the RNN model. A bidirectional LSTM network algorithm implemented the RNN architecture to extract global features, which were the inputs of the convolutional neural network. The final one-dimensional tensor represented a group of features generated through multiple convolutional and max-pooling layers. Finally, the one-dimensional tensor was fed into a fully connected layer with a sigmoid function to classify the original input URL into the fake and phishing website. The experimental results illustrated that they achieved 95.97% accuracy.

Table 5 shows the comparison of major state-of-the-art solutions. The random forest algorithm obtained higher accuracy than other models, although it varied across datasets. The UCI dataset is widely used in different machine learning models, being friendly to novices and researchers without security experience. However, it requires a process of extracting features from a URL when it is applied to real-time systems. The feature extraction process is based on security experience rules and might depend on third-party services. Many researchers proposed fusion models which combined some feature selection algorithms and a normal classifier to enhance performance and reduce the search dimensions. In terms of accuracy, deep learning-based solutions attained low performance. However, the significant advantage is that it is close to a real-time prediction system. Due to the model's input being the original URL string, it is independent of cybersecurity experience and third-party services. Once the model training is complete, the response time predicted online will be faster than traditional systems that rely on regular features.

Table 5. Comparison of major state-of-the-art solutions.

Model or Algorithm	Type	Dataset	Challenges	Limitations	Accuracy
Random forest [42]	Single	ISCXURL-2016	Achieved high accuracy and low response time without relying on third-party services and using limited features extracted from a URL.	Did not use multiple different datasets to train the model, compare the results, or to evaluate the robustness of the model.	99.57%
Random forest [45]	Single	Websites (phishTank, OpenPhish, Alexa, online payment gateway) 5223 instances: 2500 phishing URLs; 2723 legitimate URLs 20 features	The dataset is collected from the original website, 20 features are manually extracted, some features need to be obtained by calling a third-party service, and some features need to parse the website's HTML source code.	Did not use multiple different datasets to train the model, compare the results, or to evaluate the robustness of the model. The experimental dataset is small.	99.50%
PSL [1] + PART [46]	Hybrid	Websites (phishTank, Relbank) 30,500 original instances: 20,500 phishing URLs; 10,000 legitimate URLs. 3000 experiment data samples 18 features	Extracted 3000 comprehensive features and applied different set of parameters to ML models to compare the experimental results.	The legitimate URLs in the dataset are all related to banks, and some features are limited to e-banking websites.	99.30%
ISHO + SVM [47]	Hybrid	UCI	Improved spotted hyena optimization (ISHO) algorithm to select more efficient features.	The UCI dataset is open source and contains 11,055 instances with normalized features but does not contain the original URL, and the proposed approach did not contain a feature extraction procedure.	98.64%
Adaboost [48]	Single	Websites (phishTank, MillerSmiles, Google Search): size of the dataset not mentioned; each instance has 30 features	The proposed model used Weka 3.6, Python, and MATLAB [2].	Did not use multiple different datasets to train the model, compare the results, and evaluate the robustness of the model.	98.30%
LBET (logistic regression + extra tree) [4]	Hybrid	UCI	Combined meta-learning algorithms and extra trees to achieve high accuracy and low false-positive rate.	Insufficient data sources and lack of a feature extraction process.	97.57%
Bootstrap aggregating + logistic model tree [49]	Hybrid	UCI	The classifiers were trained and tested based on 10-fold cross-validation to reduce bias and variance.	Insufficient data sources and lack of a feature extraction process.	97.42%

Table 5. *Cont.*

Model or Algorithm	Type	Dataset	Challenges	Limitations	Accuracy
Random forest + neural network + bagging [26]	Hybrid	UCI	No previous research focuses on using a feedforward NN and ensemble learners for detecting phishing websites.	Insufficient data sources and lack of a feature extraction process.	97.40%
priority-based algorithms [50]	Hybrid	UCI	\	Insufficient data sources and lack of a feature extraction process.	97.00%
Random forest [51]	Single	UCI	\	Insufficient data sources and lack of a feature extraction process.	96.87%
Adam optimizer + Deep Neural Network (DNN) [52]	Deep learning	UCI	\	Insufficient data sources and lack of a feature extraction process.	96.00%
Recurrent Neural Network (RNN) + Convolutional Neural Network (CNN) [44]	Deep learning	Websites (phishTank, Alexa) 490,408 instances: 245,385 phishing URLs; 245,023 legitimate URLs features: semantic features (word embedding)	The large-scale dataset is collected from the original website. The first one to use the deep learning model to detect phishing in the context of cybersecurity issues and the first to use hundreds of thousands of phishing URLs and normal website URLs for training and testing.	The maximum length of the URL is 255 characters. Training time was too long. When the phishing website URL itself does not have relevant semantics, PDRCNN will not be able to classify correctly, and PDRCNN does not care whether the website corresponding to the URL is alive or if there is an error.	95.79%
CNN [24]	Deep learning	Websites (Alexa + OpenPhish + spamhaus.org + techhelplist.com + isc.sans.edu + phishTank) 318,642 instances. 157,626 legitimate URLs; 161,016 phishing URLs features: FG2: character embedding level features	The four different large-scale datasets are collected from original websites. The extracted features. Four different groups of features are extracted to compare the results of multiple sets of experiments.	The maximum length of the URL is 200 characters. The training time is rather long. The model is not interested in whether the URL of the website is active or has an error. The model will misclassify short links, sensitive keywords, and phishing URLs that do not imitate other websites.	95.02%
Auto encoder + NIOSELM [53]	Hybrid	Websites (phishTank, Alexa, DMOZ) 60,000 legitimate URLs; 5000 phishing URLs; 56 features	The dataset is imbalanced.	The detection accuracy may not be the best compared with the existing methods.	94.60%

Table 5. *Cont.*

Model or Algorithm	Type	Dataset	Challenges	Limitations	Accuracy
Grey wolf optimizer + SVM [54]	Hybrid	Websites (phishTank, Yahoo) 1353 instances: 805 phishing URLs; 548 legitimate URLs 30 rule-based features	It is proven that in addition to the grid search-optimized RF classifier, nature-inspired optimization algorithms can also optimize the parameters of the Support Vector Machine (SVM) model to obtain high accuracy.	The dataset is small, and there is no comparison of the results of different datasets to the model.	90.38%
Genetic algorithm (GA) + DNN [43]	Deep learning	UCI	Using GAs to select effective features and weights is a new idea.	Insufficient data sources and lack of a feature extraction process. Feature selection and weighting using GAs may require more time. The detection accuracy may be lower compared with the existing methods.	89.50%
Convolutional auto encoder + DNN [55]	Deep learning	Websites (phishTank, clients' daily requests) 6116 instance rule-based features	Features extracted based on convolutional autoencoder.	The detection accuracy may be lower compared with the existing methods. The dataset is small for deep learning models.	89.00%

[1] PSL is an abbreviation of three categories of features: Phishing Features (PF), Suspicious Features (SF), and Legitimate Features (LF).
[2] MATLAB is a programming and numeric computing platform.

6. Opportunities and Challenges

Anti-phishing techniques have been developed for decades and are improved constantly. However, there are still several challenges or limitations of phishing website detection solutions.

6.1. High-Quality Dataset

Effective phishing detection solutions should combine new data constantly for recognizing fresh rules and training machine learning models. Phishing and anti-phishing are always in the process of confronting each other. Attackers will adjust the generation of phishing links according to the published anti-phishing rules and methods. Likewise, anti-phishing needs to optimize models and algorithms based on new phishing data. Furthermore, the performance of machine learning-based solutions highly depends on the quality of the training dataset in terms of size and validation. The published datasets are small datasets that do not satisfy the demands of deep learning approaches. According to the power law, deep learning performance keeps rising with the increase of the training data size [56]. Therefore, pulling phishing URLs and legitimate URLs from websites is recommended. However, this depends on the stability of the third-party services or websites.

6.2. Efficient Features Extraction and Selection

According to published rules [22], it is not difficult to extract features from a URL. However, some rules depend on third-party services. Therefore, it might cost time and face unstable issues. Furthermore, it is important to calculate the weights of the features, decrease the dimensions, and reduce overfitting, which occurs in training processing.

Choosing the most efficient features is a matter that requires multiple computing resources, and for different models, the weighting of features may need to be adjusted.

6.3. Tiny URL Detection

Since tiny URLs do not present the real domain, resource direction, or search parameters, rule-based feature selection techniques might be useless for tiny URLs. Due to tiny URLs generated by different services, it is hard to convert them to original URLs. Furthermore, tiny URLs are short strings that are unfriendly for natural language processing to extract character-level features. If tiny URLs are not specially processed during data cleansing and preprocessing, they are likely to cause false or missed alarms. Internet products are also essential in terms of user experience, and users are also sensitive to false alarms of Internet security products.

6.4. Response Time for Real-Time Systems

Rule-based models depend on rule parsing and third-party services from a URL string. Therefore, they demand a relatively long response time in a real-time prediction system that accepts a single URL string as an input in each request from a client.

Phishing attacks spread to various communication media and target devices, such as personal computers and other smart devices. It is a big challenge for developers to cover all devices with one solution. Language independence and running environment independence should be taken into consideration to reduce system development complexity and late maintenance costs.

7. Conclusions and Future Work

This survey introduced the lifecycle of phishing to clarify the important steps for anti-phishing. This paper focuses on the technical methodologies, particularly machine learning-based solutions for phishing website detection. Furthermore, the architecture of machine learning-based resolution shows the general components in the system. The details of each part inspire the development of high-performance phishing detection techniques. We reviewed diverse academic articles and sorted diverse data sources as shown in Table 1. It is easy to start with published datasets that are standardized based on rules generated by security experts' experience. However, these datasets contain limited instances. Small datasets affect model performance in the training process, particularly for complex structured models such as multi-layer neural networks. In addition, they are relatively old, being collected approximately five years ago. The alternative method is to collect URLs from websites that contain various verified phishing URLs, such as phishtank.com. The shortcoming is that this needs an extra feature extraction process based on rules, and it depends on some third-party services. In recent years, deep learning and natural language processing techniques have developed rapidly. Some researchers saw a URL as text information and used the NLP technique to extract character-level or word-level features to feed deep learning models for predicting phishing websites. The advantage of this solution is the independence of third-party services and needless specialist experience. The disadvantage is that the learning process will cost more time.

Anti-phishing has been around for decades, and many efficient solutions have been proposed. However, attack techniques are constantly changing, and no solution is once and for all. Our continuous research of phishing website detection to defend against phishing attacks and prevent financial losses is worth it. Researchers and security experts have contributed a lot of successful resolutions, from list-based methods and rule-based strategies to machine learning-based approaches. Various machine learning-based solutions achieved higher than 95% accuracy, which is a significant advancement. However, it is believed that the accuracy performance still has space for improvement. In addition, phishing detection is sensitive to false warnings. Furthermore, a real-time system requires very low computational time. Therefore, a robust and efficient phishing website detection system still has its challenges.

Author Contributions: Writing—original draft preparation, L.T.; supervision and writing—review and editing, Q.H.M. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Johnson, J. Global Digital Population 2020. Statista. Available online: https://www.statista.com/statistics/617136/digital-population-worldwide/#:~{}:text=How%20many%20people%20use%20the (accessed on 24 July 2020).
2. 2020 Internet Crime Report. Federal Bureau of Investigation. Available online: https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf (accessed on 21 March 2021).
3. APWG. Phishing Activity Trends Report for Q4 2020. Available online: https://docs.apwg.org/reports/apwg_trends_report_q4_2020.pdf (accessed on 9 February 2021).
4. Alsariera, Y.A.; Adeyemo, V.E.; Balogun, A.O.; Alazzawi, A.K. AI Meta-Learners and Extra-Trees Algorithm for the Detection of Phishing Websites. *IEEE Access* **2020**, *8*, 142532–142542. [CrossRef]
5. Jerry, F.; Chris, H. System Security: A Hacker's Perspective. In Proceedings of the 1987 North American conference of Hewlett-Packard business computer users, Las Vegas, NV, USA, 20–25 September 1987.
6. Kumaran, N. Spam Does Not Bring Us Joy—Ridding Gmail of 100 Million More Spam Messages with TensorFlow. Google Cloud Blog. Available online: https://cloud.google.com/blog/products/g-suite/ridding-gmail-of-100-million-more-spam-messages-with-tensorflow (accessed on 6 February 2019).
7. Google Safe Browsing. Google.com. 2014. Available online: https://safebrowsing.google.com/ (accessed on 18 July 2021).
8. Basit, A.; Zafar, M.; Liu, X.; Javed, A.R.; Jalil, Z.; Kifayat, K. A comprehensive survey of AI-enabled phishing attacks detection techniques. *Telecommun. Syst.* **2020**, *76*, 139–154. [CrossRef] [PubMed]
9. Singh, C. Phishing Website Detection Based on Machine Learning: A Survey. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020. [CrossRef]
10. Vijayalakshmi, M.; Shalinie, S.M.; Yang, M.H. Web phishing detection techniques: A survey on the state-of-the-art, taxonomy and future directions. *IET Netw.* **2020**, *9*, 235–246. [CrossRef]
11. Kalaharsha, P.; Mehtre, B.M. Detecting Phishing Sites—An Overview. *arXiv* **2021**, arXiv:2103.12739.
12. Jain, A.K.; Gupta, B.B. A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterp. Inf. Syst.* **2021**, 1–39. [CrossRef]
13. Zabihimayvan, M.; Doran, D. Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection. In Proceedings of the 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, 23–26 June 2019. [CrossRef]
14. Jain, A.K.; Gupta, B.B. A novel approach to protect against phishing attacks at client side using auto-updated white-list. *EURASIP J. Inf. Secur.* **2016**. [CrossRef]
15. Tan, C.L.; Chiew, K.L.; Wong, K.; Sze, S.N. PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. *Decis. Support Syst.* **2016**, *88*, 18–27. [CrossRef]
16. Chiew, K.L.; Chang, E.H.; Sze, S.N.; Tiong, W.K. Utilisation of website logo for phishing detection. *Comput. Secur.* **2015**, *54*, 16–26. [CrossRef]
17. Mohammad, R.M.; Thabtah, F.; McCluskey, L. An Assessment of Features Related to Phishing Websites Using an Automated Technique. In Proceedings of the 2012 International Conference for Internet Technology and Secured Transactions, London, UK, 10–12 December 2012.
18. PhishTank | Join the Fight against Phishing. Available online: https://www.phishtank.com/index.php (accessed on 18 July 2021).
19. WHOIS Search, Domain Name, Website, and IP Tools—Who.is. Available online: https://who.is/ (accessed on 18 July 2021).
20. Keyword Research, Competitive Analysis, & Website Ranking | Alexa. Available online: https://www.alexa.com/ (accessed on 18 July 2021).
21. Mohammad, R.M.; Thabtah, F.; McCluskey, L. Predicting phishing websites based on self-structuring neural network. *Neural Comput. Appl.* **2013**, *25*, 443–458. [CrossRef]
22. Mohammad, R.M.A.; McCluskey, L.; Thabtah, F. UCI Machine Learning Repository: Phishing Websites Data Set. Available online: https://archive.ics.uci.edu/mL/datasets/Phishing+Websites (accessed on 26 March 2015).
23. Tan, C.L. Phishing Dataset for Machine Learning: Feature Evaluation. *Mendeley* **2018**. [CrossRef]
24. Aljofey, A.; Jiang, Q.; Qu, Q.; Huang, M.; Niyigena, J.-P. An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL. *Electronics* **2020**, *9*, 1514. [CrossRef]
25. URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. Available online: https://www.unb.ca/cic/datasets/url-2016.html (accessed on 18 July 2021).
26. Zamir, A.; Khan, H.U.; Iqbal, T.; Yousaf, N.; Aslam, F.; Anjum, A.; Hamdani, M. Phishing web site detection using diverse machine learning algorithms. *Electron. Libr.* **2020**, *38*, 65–80. [CrossRef]

27. Song, F.; Guo, Z.; Mei, D. Feature Selection Using Principal Component Analysis. *IEEE Xplore* **2010**. [CrossRef]
28. Shabudin, S.; Samsiah, N.; Akram, K.; Aliff, M. Feature Selection for Phishing Website Classification. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*. [CrossRef]
29. El-Rashidy, M.A. A Smart Model for Web Phishing Detection Based on New Proposed Feature Selection Technique. *Menoufia J. Electron. Eng. Res.* **2021**, *30*, 97–104. [CrossRef]
30. Subasi, A.; Molah, E.; Almkallawi, F.; Chaudhery, T.J. Intelligent phishing website detection using random forest classifier. In Proceedings of the 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, United Arab Emirates, 21–23 November 2017. [CrossRef]
31. Vrbančič, G.; Fister, I.; Podgorelec, V. Swarm Intelligence Approaches for Parameter Setting of Deep Learning Neural Network. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics—WIMS'18, Novi Sad, Serbia, 25–27 June 2018. [CrossRef]
32. HR, M.G.; Adithya, M.V.; Vinay, S. Development of anti-phishing browser based on random forest and rule of extraction framework. *Cybersecurity* **2020**, *3*, 20. [CrossRef]
33. Armano, G.; Marchal, S.; Asokan, N. Real-Time Client-Side Phishing Prevention Add-On. In Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), Nara, Japan, 27–30 June 2016. [CrossRef]
34. Marchal, S.; Saari, K.; Singh, N.; Asokan, N. Know Your Phish: Novel Techniques for Detecting Phishing Sites and Their Targets. In Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), Nara, Japan, 27–30 June 2016. [CrossRef]
35. Kadhim, H.Y.; Al-saedi, K.H.; Al-Hassani, M.D. Mobile Phishing Websites Detection and Prevention Using Data Mining Techniques. *Int. J. Interact. Mob. Technol. IJIM* **2019**, *13*, 205–213. [CrossRef]
36. Varjani, M.M.; Yazdian, A. PhishDetector | A True Phishing Detection System. *PhishDetector Landing Page.* Available online: https://www.moghimi.net/phishdetector (accessed on 15 July 2019).
37. Netcraft. Available online: https://www.netcraft.com/ (accessed on 7 December 2020).
38. Website Safety Check & Phishing Protection | Web of Trust. Available online: https://www.mywot.com/ (accessed on 26 May 2021).
39. Home-Pixm Anti-Phishing. Available online: https://pixm.net/ (accessed on 3 May 2021).
40. Bannister, A. Sharkcop: Google Chrome Extension Uses Machine Learning to Detect Phishing URLs. The Daily Swig | Cybersecurity News and Views. Available online: https://portswigger.net/daily-swig/sharkcop-google-chrome-extension-uses-machine-learning-to-detect-phishing-urls (accessed on 5 October 2020).
41. PhishFort Protect Anti-Phishing Cryptocurrency Browser Extension. Available online: https://www.phishfort.com/protect (accessed on 27 May 2021).
42. Gupta, B.B.; Yadav, K.; Razzak, I.; Psannis, K.; Castiglione, A.; Chang, X. A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Comput. Commun.* **2021**, *175*, 47–57. [CrossRef]
43. Ali, W.; Ahmed, A. Hybrid Intelligent Phishing Website Prediction Using Deep Neural Networks with Genetic Algorithm-based Feature Selection and Weighting. *IET Inf. Secur.* **2019**. [CrossRef]
44. Wang, W.; Zhang, F.; Luo, X.; Zhang, S. PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks. *Secur. Commun. Netw.* **2019**. [CrossRef]
45. Gandotra, E.; Gupta, D. Improving Spoofed Website Detection Using Machine Learning. *Cybern. Syst.* **2020**, *52*, 169–190. [CrossRef]
46. Barraclough, P.A.; Fehringer, G.; Woodward, J. Intelligent cyber-phishing detection for online. *Comput. Secur.* **2021**, *104*, 102123. [CrossRef]
47. Sabahno, M.; Safara, F. ISHO: Improved spotted hyena optimization algorithm for phishing website detection. *Multimed. Tools Appl.* **2021**. [CrossRef]
48. Odeh, A.; Keshta, I. PhiBoost—A novel phishing detection model Using Adaptive Boosting approach. *Jordanian J. Comput. Inf. Technol.* **2021**, *7*, 64. [CrossRef]
49. Adeyemo, V.E.; Balogun, A.O.; Mojeed, H.A.; Akande, N.O.; Adewole, K.S. Ensemble-Based Logistic Model Trees for Website Phishing Detection. *Commun. Comput. Inf. Sci.* **2021**, 627–641. [CrossRef]
50. Lakshmanarao, A.; Rao, P.; Surya, P.; Krishna, M.M.B. Phishing website detection using novel machine learning fusion approach. *IEEE Xplore* **2021**. [CrossRef]
51. Harinahalli Lokesh, G.; BoreGowda, G. Phishing website detection based on effective machine learning approach. *J. Cyber Secur. Technol.* **2020**, 1–14. [CrossRef]
52. Lakshmi, L.; Reddy, M.P.; Santhaiah, C.; Reddy, U.J. Smart Phishing Detection in Web Pages using Supervised Deep Learning Classification and Optimization Technique ADAM. *Wirel. Pers. Commun.* **2021**. [CrossRef]
53. Yang, L.; Zhang, J.; Wang, X.; Li, Z.; Li, Z.; He, Y. An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features. *Expert Syst. Appl.* **2021**, *165*, 113863. [CrossRef]

54. Anupam, S.; Kar, A.K. Phishing website detection using support vector machines and nature-inspired optimization algorithms. *Telecommun. Syst.* **2020**, *76*, 17–32. [CrossRef]
55. Deepa, S.T. Phishing Website Detection Using Novel Features and Machine Learning Approach. *Turk. J. Comput. Math. Educ. TURCOMAT* **2021**, *12*, 2648–2653. [CrossRef]
56. Mitsa, T. How Do You Know You Have Enough Training Data? Medium. Available online: https://towardsdatascience.com/how-do-you-know-you-have-enough-training-data-ad9b1fd679ee#:~{}:text=Computer%20Vision%3A%20For%20image%20classification (accessed on 23 April 2019).