Hindawi
Security and Communication Networks
Volume 2022, Article ID 5653270, 12 pages
https://doi.org/10.1155/2022/5653270

Research Article

Phishing Target Identification Based on Neural Networks Using Category Features and Images

Shihan Chen, Yixiang Lu, and Dong-Jie Liu

College of Cyber Security, Jinan University, Guangzhou 510632, Guangdong, China

Correspondence should be addressed to Dong-Jie Liu; djliu@jnu.edu.cn

Received 10 July 2022; Revised 3 November 2022; Accepted 19 November 2022; Published 6 December 2022

Academic Editor: Hamad Naeem

Phishing attack, as a significant security concern in cyberspace, has continuously threatened organizations and Internet users. For organizations, the rise in the number of phishing target brands has instilled distrust and dissatisfaction in legitimate Internet users and even damaged brand equity. Therefore, more fine-grained phishing detection mechanisms are urgently needed. In this study, we propose PTI-NN, an effective model based on neural networks that uses category features and images to identify the target brands of phishing websites. We furthermore contribute a new dataset including 3,500 phishing websites and present thirty phishing category features, which facilitate pertinent phishing detection in the field of cyber security. In the proposed PTI-NN, an embedding-based DNN is constructed to process the category features, a 2D-CNN is constructed to process the images, and finally, a fully connected layer is used to predict the target brand of phishing websites. The experimental results show that our proposed model is able to classify seventy phishing-targeted brands with a high accuracy of 91.10%, which showcases the effectiveness of our method on the identification of phishing target brands.

1. Introduction

Phishing attack, as one of the most prevalent types of cyberattacks, evolves with the rapid development of the Internet and remains as a critical security threat. In recent years, the number of phishing attacks has been on the rise. As outlined in the report published by Antiphishing Working Group (APWG), the number of phishing attacks increased substantially through 2020, roughly doubling from 2019. After that, the amount of phishing rose consistently and steadily, reaching the highest level in APWG's reporting history in December 2021. Moreover, organizations have increasingly been targets for phishing attackers that aim to steal personal property and valuable data by publishing fraudulent brand websites [1]. According to the APWG, the number of brands targeted by phishing campaigns has trended upward since 2021, peaking at 715 in September 2021. The amount of targeted brands each month in 2021 that we collect from the APWG report is shown in Figure 1. The customers' financial losses or data breaches inevitably affect organizations: it damages reputation and brand equity, meanwhile causes customer loss. Therefore, phishing detection mechanisms that more pertinently protect both organizations and individuals deserve to be of concern to security researchers.

Currently, techniques utilized for phishing detection are various, mainly classified as blacklists, heuristics, visual similarity, machine learning, and deep learning [2]. In the latest studies, researchers proposed text-based approaches using NLP algorithms to defend against phishing attacks [3]. These schemes have proven to be remarkably effective but have a common feature and a limitation that is the phishing websites' identification problem is treated as a binary classification problem. In other words, most of the work is limited to the layer of classifying suspicious websites as phishing and nonphishing. In present circumstances flooded with counterfeit brand websites, the phishing detection strategies of this layer are obviously not precise enough to achieve the purpose of pertinent prevention. This motivates us to conduct a deeper-layer research on phishing websites' identification problem.

PhishTank is a global online service that relied on crowdsourcing to collect phishing websites. Any Internet

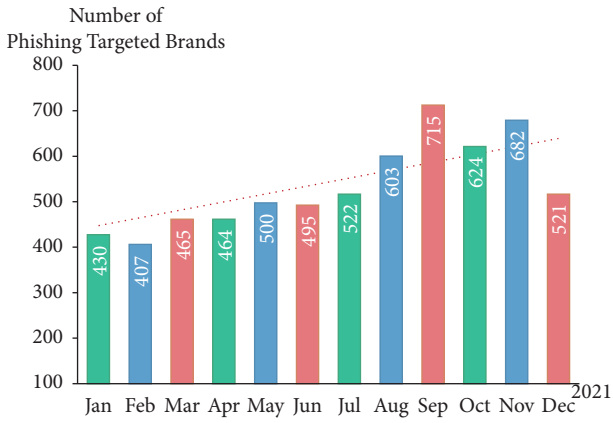

Figure 1: The number of brands targeted by phishing campaigns in 2021.

user can submit or verify phishing websites, which eventually are added to an ever-growing searchable public database [4]. Due to the accessibility of the PhishTank database, it is widely used in phishing detection research studies. In the dataset file downloaded from PhishTank, all phishing websites have a "target" label indicating the target brand of the phishing attack. However, the "target" of most phishing websites, roughly 80 percent is labeled "other," not a specific brand. This drawback goes against researchers trying to conduct more in-depth phishing analysis, such as the analysis of the distribution of target brands, the detailed statistical analysis of phishing websites targeting different brands (including the top-level domain (TLDs), the distribution of IP addresses, and the distribution of registrants), and the analysis of phishing trend. These analyses are meaningful for both organizations and individuals, contributing to the pertinent prevention of phishing attacks.

Some authoritative antiphishing organization, such as the Antiphishing Alliance of China (APAC), receives phishing reports daily and respond with the real phishing attacks as quickly as possible [5]. Reviewers are normally required to identify the target brand of phishing websites, which is a large investment of time and energy. Moreover, brands targeted by phishing attackers are numerous and have the characteristics of globality, cross-language use, and cross-regional use. Owing to the limited knowledge, reviewers are not able to accurately classify phishing websites in other languages that target nonlocal brands. Therefore, phishing disposal requires multiparty collaboration across regions and industries. A unified data-sharing format and unobstructed data-sharing channels are particularly important in relation to the effectiveness of phishing disposal [6].

For researchers in organizations or the cyber-security field, having access to phishing data identified by brand enables them to reasonably design more pertinent phishing detection mechanisms. In addition, domestic and foreign standards for phishing data sharing and blockchain-based phishing data sharing, etc., explicitly require the provision of the brands being targeted. In real world scenarios, if browsers or security plug-ins are able to provide the target brands when informing the users that they are faced with phishing attacks, which is more persuasive to them.

In light of the above statements, the fine-grained classification of phishing websites, that is, identifying the target brands of phishing websites, is exceedingly crucial. In this study, we are focused on the target identification of phishing webpages by applying different learning algorithms and different kinds of features. In the implementation of learning algorithms, the dataset and the extraction of features from dataset are always pivotal [7]. Therefore, first we construct a new dataset based on the phishing URLs of PhishTank, from which multiple category features are extracted. After that, we constructed an effective hybrid model based on neural networks, in which an embedding-based DNN processes' category features and a 2D-CNN processes images. It turns out that up to seventy target brands can be identified with a high accuracy of 91.10%. The main contributions of our work can be summarized as follows:

(i) To the best of our knowledge, we are the first to utilize the fusion of category features and images to achieve the purpose of identifying the target brands on phishing websites

(ii) Due to the full consideration of phishing characteristics, thirty features are extracted from URLs, host information, web resources, and OCR results of phishing websites

(iii) A dataset including phishing websites with their target brands, domain information, screenshots, icon pictures, and HTML contents is constructed and published for the cyber-security community [8]

(iv) A hybrid model named PTI-NN using category features and images is designed to classify the phishing brands, in which the E-DNN processes category features and the 2D-CNN processes images

2. Related Work

The related works are given in the following sections.

2.1. Phishing Target Identification. The existing studies on phishing target identification are mainly classified into two groups. One group of studies focuses on the methods of applying search engines. Wenyin et al. [9] queried keywords extracted from title information and page content on search engines and constructed a webgraph based on the websites contained in the search results. The website with the strongest parasitic relationship to the given website is regarded as the phishing target. Ramesh et al. [10] formed a set of keywords extracted from the title tag, meta tag, etc., and used Google search to discover a list of potential targets as the top results of the search. Yuan et al. [11] used the title, domain name of the URL, and web page links as query keywords for search engines to retrieve a number of phishing target candidates. Phishing target candidates that are most similar to the phishing website in title and text content are identified as the target of the phishing website. In these methods, only information from the URL and web page

content is used, so that information (e.g., brand names) from embedded objects or images cannot be exploited.

To address this limitation, Marchal et al. [12] extracted three groups of keywords from the URL, page content, and OCR result of a web page screenshot and queried them on a search engine to select the most relevant phishing target. Peng et al. [13] explored the method of using a screenshot of the web page instead of the web page content. Features extracted from the screenshot with OCR technique are treated as keywords for Google search. As an extension, Van Dooremaal et al. [14] used features extracted from the DOM and the screenshots of a web page as search terms for search engines. However, the identification accuracy of the abovementioned methods depends on the retrieval results of the search engine. The repeated network connections also consume huge runtime overhead.

Another group of studies employ methods by using a target brand list. Fu et al. [15] first proposed the scheme of using a target brand list. They calculated a similarity value between the screenshot of a suspicious website and that of all websites in the target brand list and then obtained the phishing target if the similarity value was above a threshold. Likewise, Medvet et al. [16] utilized a signature which represent the information of text and images in the web page to compare a website with all websites in the target brand list. Since the logo represents the brand of a website, Afroz and Greenstadt [17] and Wang et al. [18] explored the methods that treat the logo as an invariant to compare. They attempt to locate logos on screenshots of websites and match them with the logos in a target brand list. Along this idea, the approach to locating logos has evolved and extended to the field of computer vision. In the latest work, Lin et al. [19] used object detection to locate the logos, and similarly, tried to determine the phishing target through list comparison. However, with a large number of brands and their different variants, target brand lists need to be constantly updated in real time. In addition, the logo is not the only clue for identifying the brand of a website, for there is also other useful information (e.g., brand names) in the content of the web page.

Considering the limitations of the studies described above, we prefer to explore a method for phishing target identification with a combination of feature/information extraction and the utilization of screenshots. Meanwhile, because learning algorithms have significant performance in feature and image processing, we also prefer the methods of applying them.

2.2. Limitations of PhishTank Dataset. PhishTank provides a visual platform for the public to identify and verify phishing websites. The detailed information about each entry of suspected phishing sites are shown on the platform, including: the submission number, URL, report date, and status; a screenshot or WHOIS metadata of the website, etc. Once reported and verified, the online phishing websites will be added to a downloadable database, which is available in multiple formats and updates hourly. The downloadable database records the phishing URLs that have been reported and verified by the public, the time of report and the time of verification, and the target brand or company of the website [20].

Through our observations, the public dataset downloaded from PhishTank has two main issues from our research perspective. On one hand, only online phishing sites remain in the database. Once a phishing website transitions from online to offline status, it will no longer exist in the database. For researchers who are interested in analyzing the features or mechanisms employed by historically live and active phishing websites, it is often impossible to do so. On the other hand, very few phishing sites are labeled with brands or companies, accounting for about twenty percent, and the rest are labeled as "Other." This makes it difficult for researchers to analyze the distribution of phishing brands or the characteristics of phishing websites targeting the same brand or company.

3. Methods

This section presents our method for the detection of phishing websites. We first describe how the features are extracted from phishing websites under analysis and then introduce the proposed model for phishing target identification.

3.1. Feature Extraction. On the basis of feature definitions, we divide features into the following four categories: URL features, Host features, OCR features, and Web Resources' features. Table 1 presents the details about each category of features.

3.1.1. URL Features. Generally speaking, a normal URL consists of protocol, domain name, file path, and query parameters and so does phishing URLs. However, these components on a phishing website always behave abnormally when compared with a legitimate website [21]. Through our observation, phishing URLs attacking the same brand or company are sometimes very similar in component anomalies. Therefore, parsing the URL string itself and its components to acquire features is reliable in our work [22].

In the existing research on phishing attacks, information of phishing URL such as length and number of special symbols are often used as features [23–26]. As listed in Table 1, F1–F15 are the features related to the phishing URLs. Many phishing websites does not use HTTPS protocol but HTTP [23], thus F1 is used to denote the protocol of URL. Phishing websites always contain two or more subdomains in URL, the length of which is also quite long [24]. Therefore, F2–F5 are used to denote domain, subdomains and the depth of domain level, and F6-F7 represent the length of different part in URL. Phishing attackers tend to use special characters commonly to separate the brand name from the prefix or suffix [25], meanwhile those targeting the same brand may use the same symbol or character frequently. Thus F8–F11 are used to denote the number of numbers and special characters (such as "@," "//," "-," and ".") in the URL and F12–F13 are used to denote the char or

Table 1: Features for phishing target identification.

Features	No.	Feature identifier	Description
URL features	F1	Scheme	Scheme of URL (HTTP or HTTPS)
	F2	Domain	Domain of URL
	F3	top_domain	Top-level domain of URL
	F4	second_domain	Second-level domain of URL
	F5	domain_level	Depth of domain level
	F6	domain_len	Length of domain
	F7	behind_domain_len	Length of path
	F8	dash_count	Number of "-" in URL
	F9	num_count	Number of nums in URL
	F10	slash_count	Number of " \ " in URL
	F11	special_symbol_count	Number of "@ ~ _%#" in URL
	F12	top_char	The character appears most frequently in URL
	F13	top_symbol	The symbol appears most frequently in URL
	F14	sens_words_url	Sensitive words (i.e., "secure," "account," "login," "signing," and "confirm") in URL
	F15	url_word_top3	Top three words with the highest word frequency in URL
Host features	F16	valid_days	Valid days of domain
	F17	registrant_country	Registrant country of domain
	F18	A	IP in A record for domain (A.B.C.D)
	F19	A_1	P In A record for domain (A.B.C)
	F20	A_2	IP in A record for domain (A.B)
	F21	A_IP_num	Number of IP in A record for domain
	F22	CNAME	CNAME in CNAME record for domain
Web resources' features	F23	tag_count	Number of specific tags in Html source code (i.e., " < link > ," " < script > ," " < img > ," and " < form > ")
	F24	sens_words_html	Sensitive words (i.e., "secure," "account," "login," "signing," and "confirm") in HTML text
	F25	brand_words_html	Brand names in HTML text
	F26	tfidf_top3	Top three words with the highest tf-idf in HTML text
	F27	html_text_symbol	Number of symbols in HTML text (unicode FF00-FFEF)
	F28	icon_str	Hex string converted by.ico file
OCR features	F29	sens_words_ocr	Brand names in web page OCR results
	F30	brand_words_ocr	Sensitive words in web page OCR results

symbol appears most frequently in URL. Phishing URLs usually have multiple keywords (e.g., "secure," "account," "login," "signin," and "confirm") [26], and those targeting the same brand probably have some common words that appear frequently (e.g., brand name). Therefore, F14 is used to denote the sensitive words in URL, and F15 is used to denote the top three words with the highest word frequency.

3.1.2. Host Features. WHOIS is an Internet record listing that provides domain information about registrars, registrants, registration dates, and so on. Due to the short life cycle of phishing websites, phishing websites generally have recent registration dates, near expiration dates, or recent update date, which are provided in WHOIS database [27]. In addition, the Domain Name System (DNS) is the Internet's registry, which stores a list of domain names along with their corresponding IP addresses [28]. As commonly used DNS record type, an A-Record maps a domain name to the IP address and a CNAME Record stands for domain name aliases. Phishing attackers targeting a specific brand or company may register multiple phishing domains at the same location or use the same IP address to launch multiple phishing attacks. Therefore, querying WHOIS information and DNS records can provide many features.

Some details about host information such as domain age and registrant country are commonly used as features in research [27, 29]. As listed in Table 1, F16–F22 are the features related to WHOIS and DNS information. Domains of phishing websites targeting a specific brand may have the same valid days, which are probably registered by the same phishing attackers [29]. Therefore, F16 is used to denote the valid days of domain. For privacy protection, registrants name are not provided in WHOIS database, so registrant country is an important information which provides the location of phishing attackers [27]. F17 is used to denote the registrant country. However, for these two features are not sufficient to represent the full host information of the domain, we mined the information in the DNS record, which has not been used in the existing studies. In DNS record, A record provides IP addresses mapped from domains and CNAME record provides an alias of the domain [30]. Through our investigation, we found that some domains of phishing websites targeting a specific brand use the same IP address or alias, which is also a clue to discovering the target. Therefore, F18–F21 are used to denote the IP addresses associated with domains, and F22 is used to denote the alias of domains. It should be noted that different IP address formats represent different host areas, thus we use different features to denote them.

3.1.3. Web Resources' Features. The HTML source code of a website contains numerous HTML tags which determine how the text and images are structured and displayed in a web browser. Phishing attackers often abuse tags in HTML source code when creating phishing websites with minimal cost. Text in HTML source code often contains identity information (e.g., copyright information) related to a brand or company. Moreover, favicon is usually a unique identifying icon for brands or companies with legitimate websites. Treated as the symbol of a brand or company, favicon is generally an ICO file which contains a small image [31]. Obviously, using favicons reasonably is an important entry point to spotting features of phishing websites [28]. Therefore, useful features can be extracted from these elements in web sources.

As listed in Table 1, F23–F28 are the features related to HTML source codes and icon images. Phishing webpages tend to use external resources like scripts and images; meanwhile, forms are used to collect users' important information [32]. Therefore, F23 is used to denote the numbers of four tags including " \ link > ," " < script > ," " < img > ," and " < form > . In phishing webpages, a number of sensitive words (i.e., "secure," "account," "login," "signin," and "confirm") or brand names usually appear a number of times in the text of HTML, which are important clues to identify phishing targets. Thus, F24 and F25 are used to denote sensitive words and brand names extracted from the text of HTML. Some researchers use the TF-IDF algorithms to derive terms that mostly occur in the text of HTML, then query them as keywords in Google's search engine [33]. Inspired by this, F26 is used to denote the top three words which appear most frequently. Generally, letters in text of HTML are half-width. Through our investigation, some letters of brand words in text of HTML are replaced by full-width letters (i.e., facebook and f acebook) or special characters (i.e., amazon and α mazon). For this, F27 is used to denote the number of these symbols (Unicode FF00–FFEF) in the text of HTML. As for the favicon, we converted it into a hex string, which can represent the image. As long as phishing websites targeting a specific brand have the same favicon, the value of hex string is the same. Thus, F30 is used to denote the hex string of the favicon.

3.1.4. OCR Features. Many phishing attackers are accustomed to hiding the fraudulent information in HTML by using an obfuscation technique [13]. One of the most common techniques is that some keywords contained in the page are embedded in images or dynamically generated content by JavaScript. Applying the optical character recognition (OCR) technique to a screenshot of a web page can extract the text content of webpages, which overcomes the above obfuscations [34]. As sensitive words and brand names are important clues to identify phishing targets, extracting keywords from the OCR results can be a complement to the keywords extracted in the HTML source code. Therefore, F29 and F30 are used to denote the sensitive words and brands names extracted from the OCR results of screenshots of phishing webpages, as a complement to F24 and F25.

3.2. Model Structure. In this study, a hybrid model is proposed for processing the extracted features and images, which mainly consists of embedding-based deep neural network (E-DNN) and 2D convolutional neural network (2D-CNN). Figure 2 shows the structure of the proposed model.

DNN has great feature expression and the ability to model complex mapping, for it extracts features layer by layer and combine low-level features to form high-level features [35]. Typically, the connection of each layer in DNN is a combination of linear functions, which makes it difficult to handle sparse categorical features. In the context of neural networks, embedding is an effective way to transform those features from the original low-dimensional space to a high-dimensional space [36]. Therefore, DNN jointly with embeddings are more robust and easier to do learning on the extracted features. Among the various areas of deep learning, CNN is usually quite promising at extracting deep features from high-dimensional input such as images [37]. CNN efficiently and automatically learns the weights of the feature maps that consist of each layer, extracts abstract visual features such as points, lines, and faces from the input data, and preserves the relationships between pixels for the learning image. It has been proven to be very successful on image classification tasks [38–40].

3.2.1. E-DNN Layer. As depicted in Figure 2, E-DNN is applied to classify the extracted features. It consists of an embedding layer, a fully connected layer, and the tanh function exploited as the activation function. Let $F^i = f^i_1, f^i_2, f^i_3, \ldots, f^i_D$ be the ith sample's input features, where D is the number of feature categories. We create embedding matrices $D * M \in \mathscr{R}^{N*d}$, which indicate as an embedding layer to obtain the embedding of each feature, where N is the number of feature value categories and d is the dimension of the embedding. We then extract the input feature embedding from the following embedding matrices:

$$E^i_f = \left[M^1_{f^i_1}, M^2_{f^i_2}, \ldots, M^D_{f^i_D}\right], \tag{1}$$

where $E^i_f \in \mathscr{R}^V$ and $V = D * d$ is the concatenation of the corresponding embeddings of input features.

Then, E^i_f is fed into a fully connected layer with active function tanh. Then, the output of this layer is calculated by

$$h^i_f = \sigma\left(E^i_f \cdot W_{m1} + b_{m1}\right), \tag{2}$$

where h^i_f is the aggregated feature embedding of the ith sample, σ is the activation function, W_{m1} is the parameter matrix of fully connected Layer 1, and b_{m1} it the bias of the layer.

3.2.2. 2D-CNN Layer. As depicted in Figure 2, 2D-CNN is applied to classify the screenshots of phishing webpages. It consists of a convolutional layer, a pooling layer, two fully connected layers, and the tanh function exploited as the activation function. Let X_i be ith sample's input image. Before it is fed into the 2D-CNN layer, X_i is reshaped into

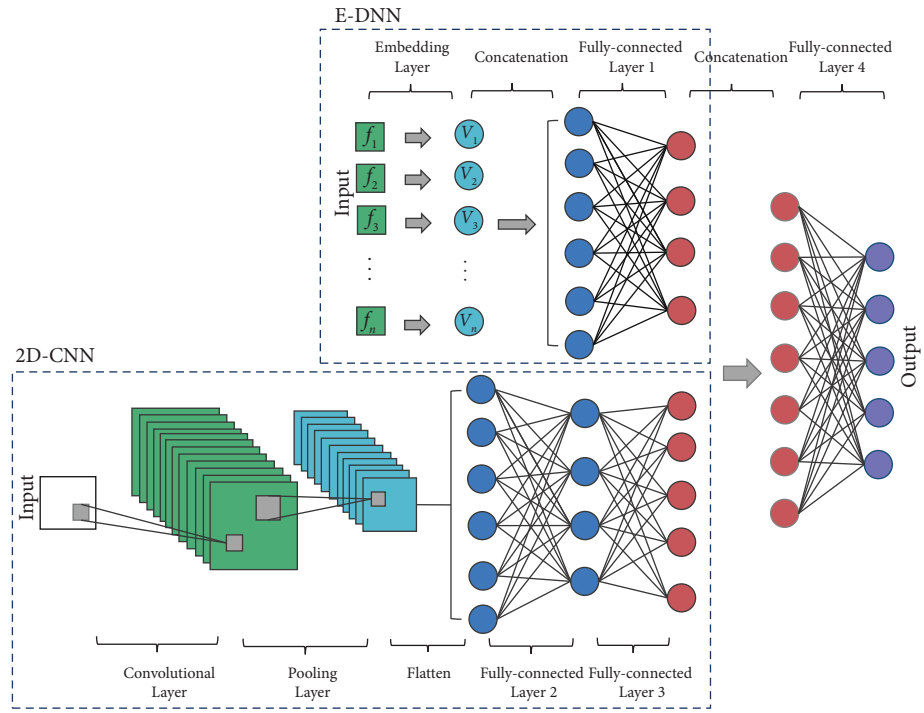

Figure 2: The structure of the proposed model.

three-channel and two-dimensional (2D) matrices. Given a convolutional layer with multiple 2D convolution kernels, each output feature map of X_i is produced by the convolution operation. Each element in the output two-dimensional convolutions is calculated as follows:

$$c_{x,y} = \sigma\left(\sum_{n=1}^{k}\sum_{m=1}^{k} w_{n,m} \cdot x_{x+n,y+m} + b\right), \tag{3}$$

where $c_{x,y}$ is the element of each output feature map, $x_{x+n,y+m}$ is the spatial position of the corresponding feature map, $w_{n,m}$ is the 2D convolution kernel with size $k \times k$, and b is the bias term.

With a pooling layer, a maxpooling operation is used to calculate the maximum value in a set of nearby inputs, which reduces total number of feature map's dimensions [41]. The maxpooling operation is expressed by

$$p_{x,y} = \max_{r \in R} c_{x \cdot t+r, y \cdot t+r}, \tag{4}$$

where $p_{x,y}$ is the pooling of feature map value $c_{x,y}$, r is the size of the pooling window, and t is the pooling stride [42].

Then, all the output feature maps of the pooling layer are flattened into a one-dimensional vector h_c^i, which is fed into two fully connected layers with active function tanh to obtain the final embedding of 2D-CNN. The calculation is expressed by

$$h_c^i = \sigma\left(\sigma\left(h_c^i W_{m2} + b_{m2}\right) W_{m2} + b_{m3}\right), \tag{5}$$

where h_c^i indicates the final embedding of the image, W_{m2} and W_{m3} indicate the parameter matrix of the fully connected Layer 2 and Layer 3, and b_{m2} and b_{m3} is the bias term of each layer.

3.2.3. Fusion. After obtaining the features from both E-DNN and 2D-CNN, we concatenate these two features into one feature vector as follows:

$$h^i = \left[h_f^{i\,T}, h_c^{i\,T}\right], \tag{6}$$

where h^i is the concatenation of the aggregated feature embedding and the image embedding which indicates the overall embedding of the sample i.

Finally, we feed h^i into the output layer with active function softmax:

$$y^i = Softmax\left(h^i W_{m4}\right), \tag{7}$$

where W_{m4} is the parameter matrix of fully connected layer 4 and $y^i \in \mathscr{R}^C$ is the prediction results of our model, where C is the number of classes and the j-dimension of y^i indicates the predicted probability of that sample i belongs to class j.

We conduct a cross entropy as the objective function:

$$\mathscr{L} = -\sum_{i=1}^{N}\sum_{j=1}^{C} t_j \log\left(f\left(X_i, F^i\right)\right), \tag{8}$$

where N is the number of samples, f indicates our model, and $(f(X_i, F^i) = y_i$ is the predicted results of our model.

4. Experiment

This section presents the procedure for dataset construction and execution of the experiment. Two groups of experiments are carried out to evaluate the effectiveness of the extracted features and the performance of the proposed model PTI-NN on our dataset. All the experiments are performed on a computer with AMD Ryzen 5 3600 6-

Core, 3.60 GHz CPU, and 8 GB RAM. Meanwhile, experimental programs are written by the Python programming language.

4.1. Dataset Construction. To address the limitations of the PhishTank database, we constructed a dataset based on the phishing websites included in the PhishTank database. Figure 3 shows the procedure for dataset construction.

4.1.1. Data Collection. The data we collected include phishing URLs, WHOIS information, DNS information, screenshots, icon images, and HTML codes. Aiming to collect as many valid phishing samples as possible, we daily perform the following steps of data collection from October 2021 to June 2022:

(i) Phishing URLs: we collect phishing websites that remained in the PhishTank database and obtain 27,403 unique URLs with submission numbers. Since our purpose is to categorize phishing brands, their corresponding targets are meanwhile transferred to our dataset.

(ii) WHOIS information: we utilize WHOIS module to look up the detailed WHOIS information of phishing websites. Necessary information, including creation date, expiration date, and registrant country are stored in our dataset.

(iii) DNS information: we parse the domain names of phishing websites to obtain DNS records through the dig command, then store the A-Records and CNAME Records of the domains in our dataset.

(iv) Screenshots: through our observation, screenshots submitted by users on the PhishTank platform have issues with inaccurate content and inconsistent size. Having accurate and uniformly formatted screenshots is important for subsequent feature extraction and model training work. Therefore, we utilize Selenium module to automatically save the screenshots of phishing webpages in an identical size and type.

(v) HTML source codes and icon images: we crawl files with an.ico extension from the host servers of phishing websites by utilizing Wget Command. The same as icon images, HTML codes are gathered by utilizing Wget Command.

4.1.2. Data Cleaning and Preprocessing. After the initial data collection, we observed that a large number of webpages were blank or not used for phishing any more. There are also some websites incorrectly associated with another brand or company. Phishing websites with these issues are considered invalid samples, which are not conducive to training effective models. So, we manually remove all the invalid phishing websites and ensure that each website is properly labeled with its corresponding brand or company. Specifically, we remove the invalid screenshots of samples and correct the labels of some samples, which ensures that the samples in our dataset could correspond to the correct target brands. As a consequence, 3,500 phishing samples are used in our experiments.

As described above, we have collected 3,500 phishing samples. As a matter of fact, the classifier will get into the problem of poor classification results when the number of samples with the same label is too small. To address this problem, we deleted the samples labeled with the same brand whose number is less than 5. For each group of experiments, we partitioned the dataset into a training set and a test set with a ratio of 8 : 2. Meanwhile, we took stratified random sampling for the samples of dataset according to the proportion of label categories and maintained this proportion in both the training set and the test set.

4.2. Model Implementation. All the models are implemented with PyTorch. For each model, we used Adam as an optimizer instead of the classical stochastic gradient descent procedure to update the network weights iteratively based on training data. The parameters of the proposed model are present in the descriptions given below.

In E-DNN, each feature is transformed into a vector of 16 dimensions in the embedding layer. All the vectors are then concatenated into a vector of size 1×960. This vector is then compressed into a vector of size 1×32 in the fully connected Layer 1. In 2D-CNN, 16 convolutional kernels of size 4×4 and stride of 4 is employed in the convolutional layer, through which each image matrix is converted into 16 matrices of size 16×16. The pooling layer conducts a maxpooling operation with a matrix of size 2×2 and a stride of 2 on the convolved 2D matrices and outputs 16 matrices of size 8×8. These matrices are then flattened into a vector of size 1×1024 which will pass through two fully connected layer. The sizes of vectors' output from the fully connected Layer 2 and the fully connected Layer 3 are 1×512 and 1×128, respectively.The vector of size 1×32 output from E-DNN and the vector of size 1×128 output from 2D-CNN are concatenated into a vector of size 1×160. Finally, the output of fully connected Layer 4 has a size for the number of labels, which is the identification for phishing target detection. In addition, we set the learning rate to 0.1, the batch size to 50, the epoch number to 100, and the random state to 0, via comparative verification.

4.3. Evaluation Metrics. To evaluate the performance of models for multiclass classification task, accuracy, macro-F1 score, and weighted-F1 score are used as assessment indicators, which are widely applied in evaluation system of multiclass classifiers. These metrics rely on the four terms of true positive (TP), true negative (TN), false negative (FN), and false positive (FP). TP and TN representing the brands of phishing samples are correctly identified by the model, while FP and FN represent the opposite of TP and TN. Different metrics can be expressed as follows.

Accuracy is the ratio of the total number of correctly classified samples to the total number of samples:

$$\text{Accuracy} = \frac{\text{TP} + \text{TF}}{\text{TP} + \text{TF} + \text{FP} + \text{FN}}. \tag{9}$$

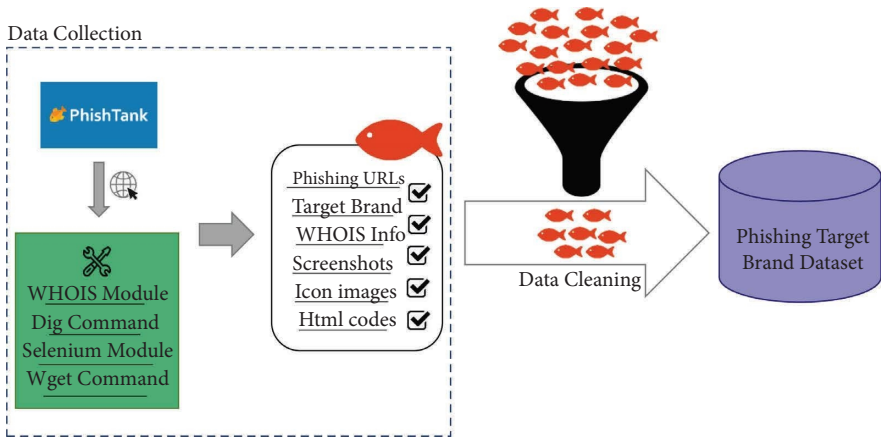

Figure 3: The procedure of dataset construction.

For each class in the samples, precision is the ratio of the numbers of correctly classified samples to the numbers of all classified samples, and recall is the ratio of the number of correctly classified samples to the numbers of samples. F1 score is the harmonic mean between precision and recall:

$$F1 - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{precision} + \text{recall}}. \tag{10}$$

Macro-F1 score and weighted-F1 score are variants of F1 score. Macro-F1 score calculates the average of F1 scores of all classes as follows:

$$\text{Macro} - F1 - \text{score} = \frac{1}{n}\sum_{i=0}^{n} F1 - \text{score}_i, \tag{11}$$

where i is the class index and n is the number of classes. On the basis of macro-F1 score, weighted-F1 score considers the problem of unbalanced distribution of class labels and performs a weighted average of F1 scores of all classes. The weight is determined based on the true distribution ratio of each class. The calculation of weighted-F1 score can be expressed by

$$\text{Weighted} - F1 - \text{score} = p_i * \sum_{i=0}^{n} F1 - \text{score}_i, \tag{12}$$

where i is the class index and p_i is the proportion of samples of class i among all the samples.

4.4. Results and Analysis

4.4.1. Evaluation of the Extracted Features. In general, the features heavily affect the performance of classifiers, especially machine learning classifiers. From this perspective, we first employ seven different machine learning models to verify whether the extracted features are effective. The seven machine learning models are SVM (support vector machine), LR (logistic regression), K-Nearest Neighbor (KNN), DT (Decision Tree), RF (Random Forest), LightGBM (light gradient boosting machine), and XGBoost (extreme gradient boosting). Considering the unbalanced sample size of each brand, accuracy, and weighted-F1 score are recognized as the main evaluation metrics. From the results presented in Table 2, the extracted features perform best in LightGBM, which eventually classifies seventy phishing target brands with an accuracy of 0.8848 and a weighted-F1 score of 0.8769. Moreover, tree-based models perform better than other nontree-based models in this experiment. This can be explained by their ability to handle categorical data with robust algorithms.

For the purpose of training an optimal model suitable for detecting phishing target brand, it is needed to select an optimal feature set. We constructed different feature sets based on the class of features, and compared their performance on a multiclass classifier. LightGBM is chosen as the classifier because it performs best on all the extracted features. Comparing the results in Tables 2 and 3, it can be seen that the feature set including all the feature extractions leads to higher classification accuracy than the other feature sets in Table 3.

To analyze whether there are redundant features, we performed a feature correlation analysis. Correlation-based feature selection can effectively reduce feature dimensionality, especially for multilabel data with multiple features [43]. The Kendall correlation coefficient is chosen for further feature correlation analysis because all the features are category features. We use the Kendall correlation coefficient to measure the correlation of each feature with the label and, meanwhile, to measure the correlation between the features. The results are shown in Figures 4 and 5, respectively. According to the result in Figure 5, the feature pairs with the values of Kendall correlation coefficient are greater than 0.8 are obtained. With the result of Figure 4, we removed five features that have the lowest correlation with the label. The retained twenty five features are fed into seven machine learning models, and the result is shown in Table 4. From the results of Tables 2 and 4, it is observed that the accuracy and weighted-F1 scores of the models using twenty five features were lower than those of the models using thirty features. Therefore, the thirty extracted features are verified as the optimal feature set.

Table 2: The performance of extracted features on seven machine learning classifiers.

Models	Accuracy	Macro-F1 score	Weighted-F1 score
SVM	0.3699	0.0459	0.2594
LR	0.4293	0.1343	0.3648
KNN	0.6318	0.3499	0.6042
DT	0.8394	0.6611	0.8319
RF	0.8586	**0.7236**	0.8429
XGBoost	0.8743	0.7075	0.8677
LightGBM	**0.8848**	0.7172	**0.8769**

The bold values represent the best values of the evaluation metrics.

Table 3: The performance of different features' sets on LightGBM.

Feature set	Accuracy	Macro-F1 score	Weighted-F1 score
$f_U + f_H + f_W$	**0.8325**	**0.6655**	**0.8197**
$f_U + f_H + f_O$	0.8743	0.7005	0.8659
$f_U + f_W + f_O$	0.8778	0.7115	0.8692
$f_H + f_W + f_O$	0.8813	0.7313	0.8768
$f_U + f_H$	0.7958	0.6375	0.7771
$f_U + f_W$	0.8255	0.6647	0.8103
$f_U + f_O$	0.8674	0.6829	0.8581
$f_H + f_W$	0.7941	0.6576	0.7774
$f_H + f_O$	0.8743	0.7115	0.8680
$f_W + f_O$	0.8551	0.7415	0.8521
f_U	0.7644	0.5805	0.7418
f_H	0.7365	0.5838	0.7104
f_W	0.6143	0.5067	0.5849
f_O	0.2583	0.0064	0.1172

f_U represents URL features, f_H represents host features, f_W represents web resource features, and f_O represents OCR features. The bold values represent the best values of the evaluation metrics.

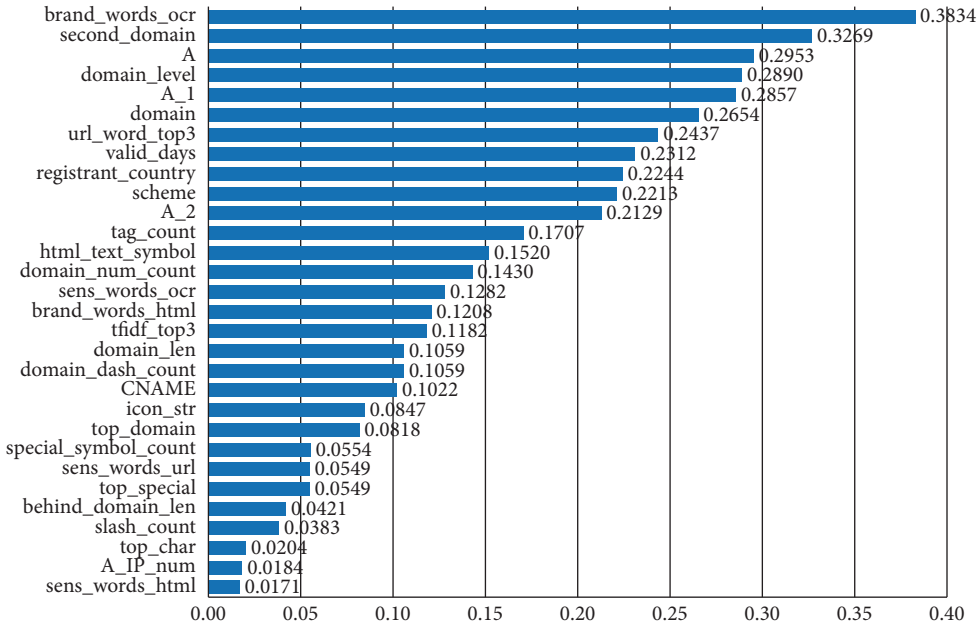

Figure 4: Correlation between the extracted features and label.

4.4.2. Evaluation of Models. To obtain an optimal classifier that handles feature set, we evaluate the performance of deep neural network classifiers. Table 5 reports the results for DNN and embedding-based DNN (E-DNN) when the optimal feature set is used. From the results of Tables 2 and 5, we observe that E-DNN classifier outperforms the other classifier, including seven machine learning classifiers and DNN, having the highest accuracy of 0.8918 and weighted-F1 score of 0.8773. This is because embedding is an effective way to transform those features from the original low-

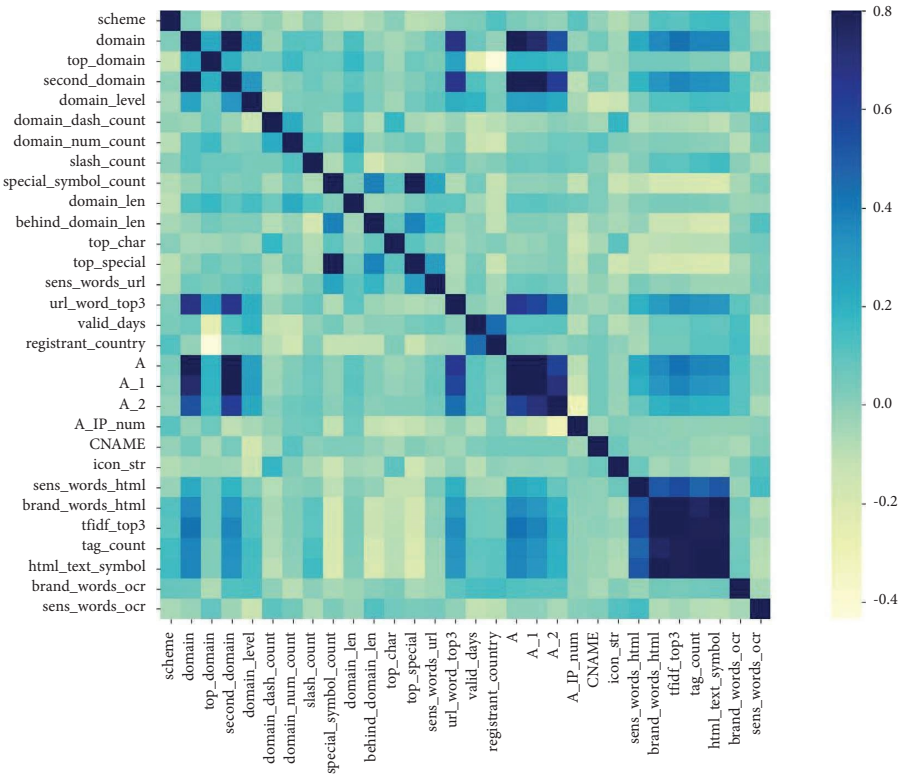

Figure 5: Correlation between the extracted features.

Table 4: The performance of models based on feature correlation results.

Models	Accuracy	Macro-F1 score	Weighted-F1 score
SVM	0.3717	0.0422	0.2557
LR	0.4119	0.1312	0.3487
KNN	0.6300	0.3544	0.6074
DT	0.8429	0.6614	0.8367
RF	0.8556	0.7306	0.8533
XGBoost	**0.8796**	**0.7317**	**0.8722**
LightGBM	0.8743	0.7120	0.8646

The bold values represent the best values of the evaluation metrics.

Table 5: The performance of deep learning classifiers using the optimal feature set.

Models	Accuracy	Macro-F1 score	Weighted-F1 score
DNN	0.8539	0.7095	0.8513
E-DNN	**0.8918**	**0.7518**	**0.8773**

The bold values represent the best values of the evaluation metrics.

dimensional space to a high-dimensional space, which reduce redundant information in the features.

The performance of E-DNN using only feature sets, 2D-CNN using only images, and the hybrid model PTI-NN using both feature sets and images are evaluated for obtaining an optimal model for this phishing target identification task. As the experimental results presented in Table 6, PTI-NN is the optimal model that achieves 91.10% accuracy. E-DNN and 2D-CNN also perform well, with an accuracy of 89.18% and 88.13%, respectively. This means the model that uses both feature set and images has a higher accuracy on this multiclass classification task than those models that only use feature data or image data. Therefore, the hybrid model is the optimal model in this study, which meanwhile showcases the effectiveness of our method on the phishing target identification task.

Table 6: The performance results of our proposed models: E-DNN, 2D-CNN, and PTI-NN.

Models	Accuracy	Macro-F1 score	Weighted-F1 score
E-DNN	0.8918	0.7518	0.8773
2D-CNN	0.8813	0.7536	0.8733
PTI-NN	**0.9110**	**0.7860**	**0.9050**

The bold values represent the best values of the evaluation metrics.

5. Conclusion

As brands are increasingly attacked by phishing groups, in this study, we propose a hybrid model named PTI-NN to identify the target brands of phishing websites. We construct a new dataset and overcome the limitations of the PhishTank database. We then use the features extracted from our dataset and the screenshots in the dataset as input to the models. In evaluation experiments, seven machine learning models are implemented. According to the experimental results, all the tree models and E-DNN which take the extracted features as input are able to classify seventy brands with high accuracy. Meanwhile, E-DNN outperformed machine learning models. Using both category features and

images, PTI-NN, which consists of E-DNN and 2D-CNN, is the optimal model for this task.

In future work, we intend to adjust the model structure to promote the classification result. For example, parameter optimization techniques can be used to tune model parameters, and more efficient features or image processing models can be implemented.

Data Availability

The experimental dataset has been published in the open source community, and the link to access it can be found in reference [8].

Conflicts of Interest

The authors declared they have no conflicts of interest for this research.

Authors' Contributions

Shihan Chen and Yixiang Lu contributed equally to this work.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 92067108, Natural Science Foundation of Guangdong Province under Grant No. 2022A050520013, and Macau Science and Technology Development Funds under Grant No. 0059/2021/AGJ.

References

[1] A. Alhogail and A. Alsabih, "Applying machine learning and natural language processing to detect phishing email," *Computers & Security*, vol. 110, Article ID 102414, 2021.

[2] G. Varshney, M. Misra, and P. K. Atrey, "A survey and classification of web phishing detection schemes," *Security and Communication Networks*, vol. 9, no. 18, pp. 6266–6284, 2016.

[3] R. A. A. Jonker, R. Poudel, T. Pedrosa, and R. P. Lopes, "Using natural language processing for phishing detection," in *Proceedings of the International Conference on Optimization*, pp. 540–552, Germany, 2021.

[4] D. G. Dobolyi and A. Abbasi, "Phishmonger: a free and open source public archive of real-world phishing websites," in *Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics*, pp. 31–36, San Antonio, TX, USA, 2016.

[5] H. Bo, W. Wei, W. Liming et al., "A hybrid system to find & fight phishing attacks actively," in *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1, pp. 506–509, IEEE, 2011.

[6] D. Liu, W. Wang, Y. Wang, and Y. Tan, "Phishledger: a decentralized phishing data sharing mechanism," in *Proceedings of the 2019 International Electronics Communication Conference*, pp. 84–89, Japan, 2019.

[7] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.

[8] S. Chen, Y. Lu, and D.-J. Liu, "Phishing target dataset," 2022, https://github.com/yatpit/Phishing-Target-Dataset.git.

[9] L. Wenyin, G. Liu, B. Qiu, and X. Quan, "Antiphishing through phishing target discovery," *IEEE Internet Computing*, vol. 16, no. 2, pp. 52–61, 2012.

[10] G. Ramesh, I. Krishnamurthi, and K. S. S. Kumar, "An efficacious method for detecting phishing webpages through target domain identification," *Decision Support Systems*, vol. 61, pp. 12–22, 2014.

[11] H. Yuan, X. Chen, Y. Li, Z. Yang, and W. Liu, "Detecting phishing websites and targets based on URLs and webpage links," in *Proceedings of the 2018 24th International Conference on Pattern Recognition*, pp. 3669–3674, Beijing, China, 2018.

[12] S. Marchal, K. Saari, N. Singh, and N. Asokan, "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*, pp. 323–333, ICDCS) IEEE, Japan, 2016.

[13] P. Peng, C. Xu, L. Quinn, H. Hu, B. Viswanath, and G. Wang, "What happens after you leak your password: understanding credential sharing on phishing sites," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pp. 181–192, Auckland, New Zealand, 2019.

[14] B. Van Dooremaal, P. Burda, L. Allodi, and N. Zannone, "Combining text and visual features to improve the identification of cloned webpages for early phishing detection," in *Proceedings of the 16th International Conference on Availability*, pp. 1–10, Reliability and Security, Portugal, 2021.

[15] A. Y. Fu, L. Wenyin, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD)," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 301–311, 2006.

[16] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks*, pp. 1–6, Washington, DC, USA, 2008.

[17] S. Afroz and R. Greenstadt, "Phishzoo: Detecting Phishing Websites by Looking at Them," in *Proceedings of the 2011 IEEE fifth international conference on semantic computing IEEE*, pp. 368–375, Palo Alto, CA, USA, 2011.

[18] G. Wang, H. Liu, S. Becerra et al., *Verilogo: Proactive Phishing Detection via Logo Recognition*, Department of Computer Science and Engineering, University of California, California, CA, USA, 2011.

[19] Y. Lin, R. Liu, D. M. Divakaran et al., "Phishpedia: a hybrid deep learning based approach to visually identify phishing webpages," *30th USENIX Security Symposium*, vol. 21, pp. 3793–3810, 2021.

[20] T. Moore and R. Clayton, "Evaluating the wisdom of crowds in assessing phishing websites," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 16–30, Springer, Berlin, Heidelberg, 2008.

[21] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu, "OFS-NN: an effective phishing websites detection model based on optimal feature selection and neural network," *IEEE Access*, vol. 7, pp. 73271–73284, 2019.

[22] L. Yang, J. Zhang, X. Wang, Z. Li, Z. Li, and Y. He, "An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features," *Expert Systems with Applications*, vol. 165, Article ID 113863, 2021.

[23] A. K. Jain and B. Gupta, *PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning*, pp. 467–474, Cyber Security Springer, Singapore, 2018.

[24] A. K. Jain and B. Gupta, "Comparative analysis of features based machine learning approaches for phishing detection," in *Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development*, pp. 2125–2130, INDIACom) IEEE, New Delhi, India, 2016.

[25] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule-based phishing websites classification," *IET Information Security*, vol. 8, no. 3, pp. 153–160, 2014.

[26] J. Hong, T. Kim, J. Liu, N. Park, and S. W. Kim, "Phishing url detection with lexical features and blacklisted domains," in *Adaptive Autonomous Secure Cyber Systems*, pp. 253–267, Springer, Germany, 2020.

[27] K. Althobaiti, G. Rummani, and K. Vaniea, "A review of human-and computer-facing URL phishing features," in *Proceedings of the 2019 IEEE European Symposium on Security and Privacy Workshops*, pp. 182–191, EuroS&PW) IEEE, Italy, 2019.

[28] G. Sonowal, "Communication channels," in *Phishing and Communication Channels*, pp. 51–75, Springer, Germany, 2022.

[29] E. Zhu, C. Ye, D. Liu, F. Liu, F. Wang, and X. Li, "An effective neural network phishing detection model based on optimal feature selection," in *Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing*, pp. 781–787, Melbourne, Australia, December 2018.

[30] S. Hao, N. Feamster, and R. Pandrangi, "Monitoring the initial DNS behavior of malicious domains," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, pp. 269–278, Berlin, 2011.

[31] G. G. Geng, X. D. Lee, and Y. M. Zhang, "Combating phishing attacks via brand identity and authorization features," *Security and Communication Networks*, vol. 8, no. 6, pp. 888–898, 2015.

[32] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, pp. 27–39, 2019.

[33] H. H. Nguyen and D. T. Nguyen, "Machine learning based phishing web sites detection," in *AETA 2015: Recent Advances in Electrical Engineering and Related Sciences*, pp. 123–131, Springer, Germany, 2016.

[34] K. Tian, S. T. Jan, H. Hu, D. Yao, and G. Wang, "Needle in a haystack: tracking down elite phishing domains in the wild," in *Proceedings of the Internet Measurement Conference 2018*, pp. 429–442, Boston, MA, USA, 2018.

[35] H. Yi, S. Shiyu, D. Xiusheng, and C. Zhigang, "A study on deep neural networks framework," in *Proceedings of the 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference*, pp. 1519–1522, IMCEC) IEEE, China, 2016.

[36] E. Golinko and X. Zhu, "Generalized feature embedding for supervised, unsupervised, and online learning tasks," *Information Systems Frontiers*, vol. 21, no. 1, pp. 125–142, 2019.

[37] H. Naeem and A. A. Bin-Salem, "A CNN-LSTM network with multi-level feature extraction-based approach for automated detection of coronavirus from CT scan and X-ray images," *Applied Soft Computing*, vol. 113, Article ID 107918, 2021.

[38] Z. Yan, H. Zhang, R. Piramuthu et al., "Hierarchical deep convolutional neural networks for large scale visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2740–2748, Italy, 2015.

[39] X. Lei, H. Pan, and X. Huang, "A dilated CNN model for image classification," *IEEE Access*, vol. 7, pp. 124087–124095, 2019.

[40] K. Liu, J. Li, and S. S. Hussain Bukhari, *Overview of Image Inpainting and Forensic Technology*, p. 2022, Security and Communication Networks, London, UK, 2022.

[41] H. Naeem, F. Ullah, M. R. Naeem et al., "Malware detection in industrial internet of things based on hybrid image visualization and deep learning model," *Ad Hoc Networks*, vol. 105, Article ID 102154, 2020.

[42] C. Chen, Z. Hua, R. Zhang, G. Liu, and W. Wen, "Automated arrhythmia classification based on a combination network of CNN and LSTM," *Biomedical Signal Processing and Control*, vol. 57, Article ID 101819, 2020.

[43] L. Jiang, G. Yu, M. Guo, and J. Wang, "Feature selection with missing labels based on label compression and local feature correlation," *Neurocomputing*, vol. 395, pp. 95–106, 2020.